STUDY ON ENHANCEMENT OF DOMINATING SETS IN AD HOC WIRELESS NETWORKS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

in

Computer Science and Engineering

By

S. Jayadev

Department of Computer Science and Engineering

National Institute of Technology, Rourkela

Rourkela, Orissa 769008, India

May 2007

STUDY ON ENHANCEMENT OF DOMINATING SETS IN AD HOC WIRELESS NETWORKS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

in

Computer Science and Engineering

By

S. Jayadev

Under guidence of

Prof. S. K. Rath

Department of Computer Science and Engineering

National Institute of Technology, Rourkela

Rourkela, Orissa 769008, India

May 2007

National Institute of Technology, Rourkela

Orissa, 769008, India.

CERTIFICATE

This is to certify that the Thesis entitled "**Study on Enhancement of Dominating Sets in Ad Hoc Wireless Networks**" submitted by Sri **S. Jayadev** in partial fulfillment of the requirements for the award of Master of Technology Degree in Computer Science and Engineering with specialization in "Computer Science and Engineering" at the National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/ Institute for the award of any degree or diploma.

Date: May 2007

Prof. S. K. Rath
Department of Computer Science and Engg
National Institute of Technology
Rourkela -769008

ACKNOWLEDGEMENTS

No thesis is created entirely by an individual, many people have helped to create this thesis and each of their contribution has been valuable. I express my sincere gratitude to my thesis supervisor, Dr. S. K. Rath, Professor, CSE, for his kind and valuable guidance for the completion of the thesis work. His consistent support and intellectual guidance made me energize and innovate new ideas. I am grateful to Dr. S. K. Jena, Professor and Head, CSE for his excellent support during my work. I am also thankful to Prof B. Majhi, Dr. D. P. Mahopatra, Dr A. K. Turuk, Dr. R. Baliarsingh,Mr. B. D. Sahoo of CSE Department, for providing me support and advice in preparing my thesis work. Thanks to all my classmates for their love and support. Last, but not least I would like to thank my parents for supporting me to do complete my masters degree in all ways.

S. Jayadev

Roll No: 20506007

M.Tech, Computer Science and Engineering

N.I.T, Rourkela - 769008.

ABSTRACT

An essential component of effective use of Ad Hoc Wireless networks is proper utilization of available resources and network stability. There has been a recent increase of interest in Ad Hoc networks, partly due to the fact that Ad Hoc networks can be implemented without requiring any support from existing wired backbone or costly hardware setup.

The effective utilization of the power of the Ad Hoc networks lies in scalability, stability, maintainability and rapid convergence of these networks. The route detection and route formation in the Ad Hoc networks should be done instantaneously. Performing effective route detection on such environments can be best achieved when the broadcasting is improved and the redundancy in route detection is removed. In this thesis work, a cluster based algorithm as well as a localized algorithm for Connected Dominating Set formation have been proposed.

The Clustering algorithm forms disjoint logical groups of nodes with a lead node as head of the cluster. The cluster head takes part in the routing and has additional tasks of cluster maintenance. The clusters form a logical backbone and the Route search space is limited to the no. of clusters. But, the method forms suboptimal routes.

The Connected Dominating Set based algorithm forms a logical backbone of connected gateways. The broadcast packets are retransmitted by these nodes. The routes are computed by shortest path algorithm. Changes in the backbone does not have impact on the ongoing communications. New routes are computed upon failure of existing routes.

The results obtained using this algorithm are compared with other methods across a range of different scenarios with 100 nodes upon a simulation area of 100 X 100 twips. The Connected Dominating Set algorithm distributes the dominating nodes equally along the entire range while others are being biased towards higher “id” nodes. The Connected Dominating Set based algorithm is excelling the cluster algorithm in Search space size.

Abbreviations

CDS	Connected Dominating Set
DCA	Weight Based Distributed Clustering Algorithm
SPA	Weight Based Self Pruning Algorithm
MCDS	Minimum Connected Dominating Sets
WCA	Weight Based Clustering Algorithm
DMAC	Distributed Mobility Adaptive Clustering Algorithm
DSDV	Destination-Sequenced Distance-Vector Routing
WRP	Wireless Routing Protocol
OLSR	Optimized Link State Routing
DSR	Dynamic Source Routing
AODV	Ad hoc On-demand Distance Vector
TORA	Temporally Ordered Routing Algorithm
DG	Distributed Gateway
GA	Genetic Algorithms
MN	Mobile Node
DAG	Directed Acyclic Graph
GPS	Global Positioning System
CGSR	Cluster Head Gateway Switch Routing Protocol
STAR	Source Tree Adaptive Routing
FSR	Fisheye State Routing
HSR	Hierarchical State Routing

GSR	Global State Routing
ABR	Associativity Based Routing
SSA	Signal Stability Based Adaptive Routing
FORP	Flow Oriented Routing Protocol
PLBR	Preferred Link Based Routing
CEDAR	Core Extraction Distributed Ad Hoc Routing
ZRP	Zone Routing Protocol
ZHLS	Zone Based Hierarchical link State Routing
RIP	Routing Information Protocol
OSPF	Open Shortest Path First
MPR	Multi Point Relay
DARPA	Defense Advanced Research Projects Agency
RREQ	Route Request
RREP	Route Reply

Contents

List of Figures

Chapter 1

Introduction

In this new era of communications, mobile communication and mobile computing have exhibited a tremendous rise in popularity among reasearchers and practitioners. The advent of new powerful, efficient and compact devices like Personnel Digital Assistants (PDAs), Pagers, Laptops, Cellular Phones, having extraordinary processing power paved the way for better communication technologies and mobile computations. At the same time, market for the wireless telephones and the Communication devices are experiencing a rapid growth. The availability of Internet and the Internet based applications in these devices delivered through emerging technologies is giving a push to the move.

The thirst for information is increasing day to day. The objective of getting right quality information is becoming the key point of interest whether you are at home, office or on the go. The demand for use of right methodology to get right quality information is rising in a very fast pace. The multimedia content provided over these devices is pushing up their popularity. The availability on the go of information implies that communication devices need to support user mobility. This aspect invites a new dimension to solutions of problems in this domain.

Mobility causes the unpredictable resource requirements and problems in connectivity. Solutions to these problems have boosted the market for the wireless devices. The services like voice communication, Internet, and video on demand, e.t.c., traditionally supported over wired networks are currently being supported over mobile networks.

There are several problems that wireless networks face today these include intermittent service, limited resources (such as Bandwidth, limited Battery Power) while providing the user mobility. The researchers are trying hard to get out of these problems and improve the quality of service in the wireless networks.

Wireless networks that are discussed solely rely on the wired backbone by which all base stations are connected, implying that networks are fixed and constrained to a geographical area with a pre-defined boundary. Mobility of nodes in the network is allowed up to a predefined geographical boundary at which the network is deployed. Deployment of such networks takes time and cannot be set up in times of utmost emergency.

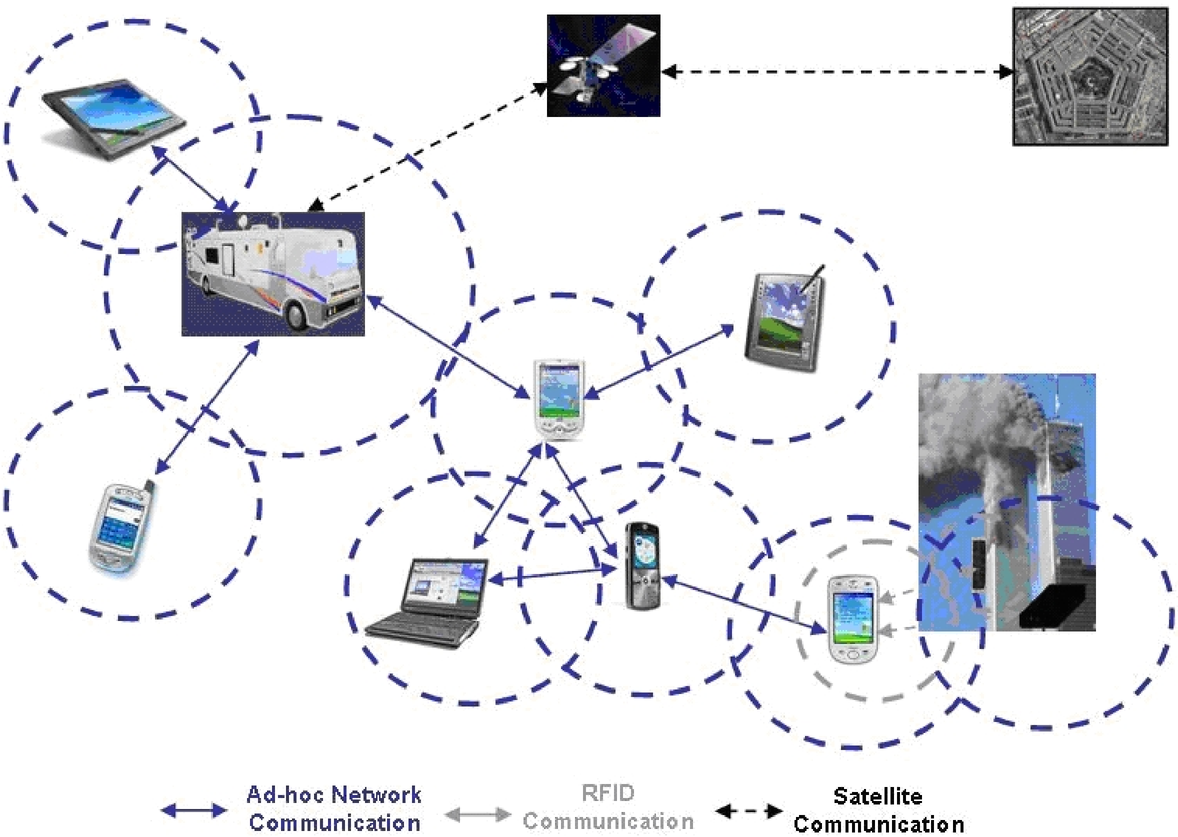


Figure 1.1: Typical Ad Hoc Network

1.1 Ad hoc wireless networks

The term mobile Ad Hoc networks describes distributed, mobile, wireless, multi-hop networks that operate without any benefit from the existing fixed infrastructure except for the nodes themselves. The concept of mobile packet radio networks, where every node in the network is mobile and where wireless multi-hop (store and forward) routing is utilized ways back to 1970s [11], not long after the initial development packet switching technology that grew into what we know as Internet, the U.S. Department of Defense sponsored research to enable packet switching technology to operate without the restrictions of fixed or wired infrastructure.

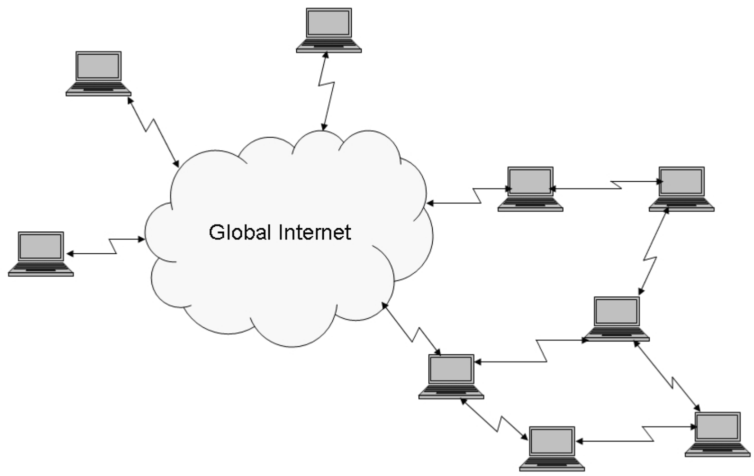


Figure 1.2: Ad Hoc Network Connected to Internet

The recent technical advancements made these communication devices compact and cheaper while being more robust. This inturn leads to newer applications like personnel area communications for resource sharing and vehicular communications for priory route detection and driver assistance.

The cellular networks operate on single-hop wireless links and the communications are supported by wired backbones. The operations are controlled by a centralized coordinator which makes routing decisions. The task of the mobile nodes is to maintain the connections with base station and to have single hop wireless transmissions. All the routing algorithms used for wired networks can also be applied to cellular networks except with certain exceptions like provision for call handoff. Ad Hoc networks require more dynamic and robust algorithms to handle the mobility of the nodes

1.2 Motivation for Thesis

Routing algorithms in Ad Hoc Networks are more challenging in nature. Economical deployment of Ad Hoc networks requires efficient routing algorithms. Many Routing Algorithms work properly in less denser networks and their performance is compromised in dense networks. This is due to their dependence on broadcasting for route identification. Improving the efficiency of route identification requires a virtual sparse forwarding network. Clustering and Connected Dominating Sets form the major solutions to the problem.

In clustering, Cluster Heads form a virtual backbone while other nodes in the network associate with these cluster heads to form virtual clusters. Cluster heads are elected or selected based on mutual consent. Clustering comes with additional communication overhead for identification of gateways. Connected dominating sets identify a connected set of gateways, uses it in broadcasting.

1.3 Organization of Thesis

This thesis is divided into six Chapters. The Chapter 1, that is here, gives some introduction and motivation for concentration on Dominating sets. Chapter 2 presents the Mobility Models used for protocol evaluation in Ad Hoc Wireless networks. Chapter 3 presents classification of existing Routing Algorithms. Chapter 4 presents the cluster based dominating sets and proposes Weight Based Distributed Clustering algorithm(DCA). Chapter 5, presents the Connected Dominating sets and proposed Weight Based Self Pruning Algorithm(SPA). And the last chapter presents the conclusion and proposals for possible extensions of the thesis work.

Chapter 2

Mobility Patterns

2.1 Introduction

In order to thoroughly simulate a new protocol for an ad hoc network, it is imperative to use a mobility model that accurately represents the mobile nodes (MNs) that will eventually utilize the given protocol. Only in this type of scenario is it possible to determine whether or not the proposed protocol will be useful when implemented. Currently there are two types of mobility models used in the simulation of networks: traces and synthetic models [22]. Traces are those mobility patterns that are observed in real life systems. Traces provide accurate information, especially when they involve a large number of participants and an appropriately long observation period. However, new network environments (e.g. ad hoc networks) are not easily modeled if traces have not yet been created. In this type of situation it is necessary to use synthetic models. Synthetic models attempt to realistically represent the behaviors of MNs without the use of traces. This thesis presents several synthetic mobility models that have been proposed for (or used in) the performance evaluation of ad hoc network protocols.

A mobility model should attempt to mimic the movements of real MNs. Changes in speed and direction must occur and they must occur in reasonable time slots. For example, MNs cannot be constrained to travel in straight lines at constant speeds throughout the course of the entire simulation because real MNs would not travel in such a restricted manner.

This chapter discusses three different synthetic entity mobility models for ad hoc networks:

1. Random Walk Mobility Model (including its many derivatives): A simple mobility model based on random directions and speeds.

2. Random Waypoint Mobility Model: A model that includes pause times between changes in destination and speed.

3. Random Direction Mobility Model: A model that forces MNs to travel to the edge of the simulation area before changing direction and speed

2.2 Random Walk

The Random Walk Mobility Model was first described mathematically by Einstein in 1926 [24]. Since many entities in nature move in extremely unpredictable ways, the Random Walk Mobility Model was developed to mimic this erratic movement [23]. In this mobility model, an MN moves from its current location to a new location by randomly choosing a direction and speed in which to travel. The new speed and direction are both chosen from pre-defined ranges, [speedmin; speedmax] and $[0; 2\pi]$ respectively. Each movement in the Random Walk Mobility Model occurs in either a constant time interval t or a constant distance traveled d, at the end of which a new direction and speed are calculated. If an MN which moves according to this model reaches a simulation boundary, it “bounces ”off the simulation border with an angle determined by the incoming direction. The MN then continues along this new path.

Many derivatives of the Random Walk Mobility Model have been developed including the 1-D, 2-D, 3-D, and d-D walks. In 1921, Polya proved that a random walk on a one or two-dimensional surface returns to the origin with complete certainty, i.e., a probability of 1.0 [25]. This characteristic ensures that the random walk represents a mobility model that tests the movements of entities around their starting points, without worry of the entities wandering away never to return.

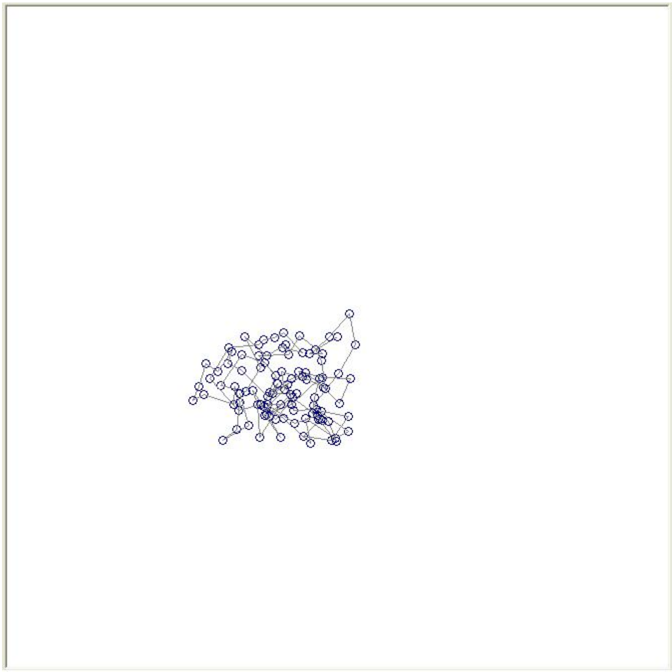


Figure 2.1: Random Walk mobility Model

The 2-D Random Walk Mobility Model is of special interest, since the Earth's surface is modeled using a 2-D representation. Figure 2.2 shows an example of the movement observed from this 2-D model. The MN begins its movement in the center of the 300mx300m simulation area or position (50, 50). At each point, the MN randomly chooses a direction between 0 and 2π and a speed between 0 and 10 m/s. The MN is allowed to travel for 60 seconds before changing direction and speed. In the Random Walk Mobility Model, an MN may change direction after traveling a specified distance instead of a specified time. In this example, the MN travels for a total of 10 steps (instead of 60 seconds) before changing its direction and speed. Unlike Figure 2.1, each movement of the MN in Figure 2.2 is the exact same distance.

The Random Walk Mobility Model is a widely used mobility model which is sometimes referred to as Brownian Motion. In its use the model is sometimes simplified

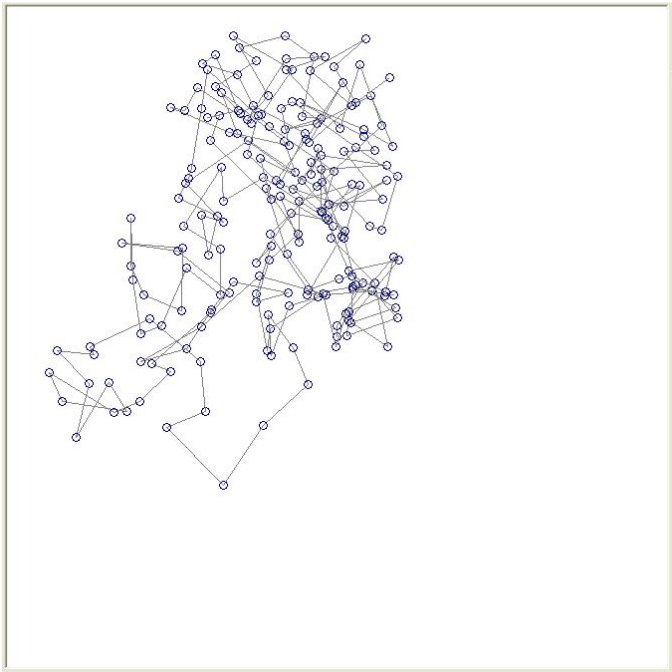


Figure 2.2: Random Walk mobility Model

2.3 Random Waypoint

The Random Waypoint Mobility Model includes pause times between changes in direction and/or speed [26]. An MN begins by staying in one location for a certain period of time (i.e., a pause time). Once this time expires, the MN chooses a random destination in the simulation area and a speed that is uniformly distributed between [Minspeed, Maxspeed]. The MN then travels toward the newly chosen destination at the selected speed. Upon arrival, the MN pauses for a specified time period before starting the process again.

Figure 2.3 shows an example traveling pattern of an MN using the Random Waypoint Mobility Model starting at a randomly chosen point or position (133, 180); the speed of the MN in the figure is uniformly chosen between 0 and 10 m/s. The movement pattern of an MN using the Random Waypoint Mobility Model is similar to the Random Walk Mobility Model if pause time is zero and [minspeed, maxspeed] = [speedmin, speedmax].

The Random Waypoint Mobility Model is also a widely used mobility model. In addition, the model is sometimes simplified.

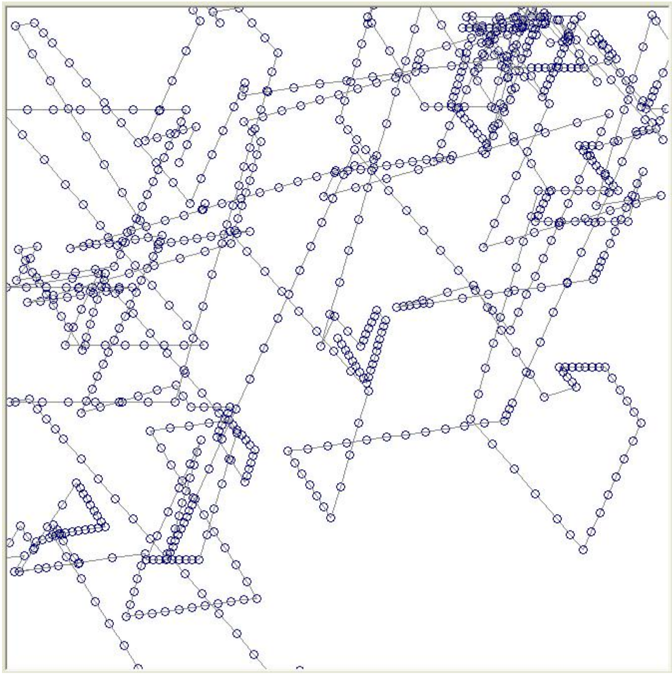


Figure 2.3: Random Waypoint mobility Model

2.4 Random Direction

The Random Direction Mobility Model [47] was created to overcome density waves in the average number of neighbors produced by the Random Waypoint Mobility Model. A density wave is the clustering of nodes in one part of the simulation area. In the case of the Random Waypoint Mobility Model, this clustering occurs near the center of the simulation area. In the Random Waypoint Mobility Model, the probability of an MN choosing a new destination that is located in the center of the simulation area, or a destination which requires travel through the middle of the simulation area, is high. Thus, the MNs appear to converge, disperse, and converge again.

In order to alleviate this type of behavior and promote a semi-constant number of neighbors throughout the simulation, the Random Direction Mobility Model was developed [47]. In this model, MNs choose a random direction in which to travel similar to the Random Walk Mobility Model. An MN then travels to the border of the simulation area in that direction.Once the simulation boundary is reached, the MN pauses for a specified time, chooses another angular direction (between 0 and 180 degrees) and continues the process.

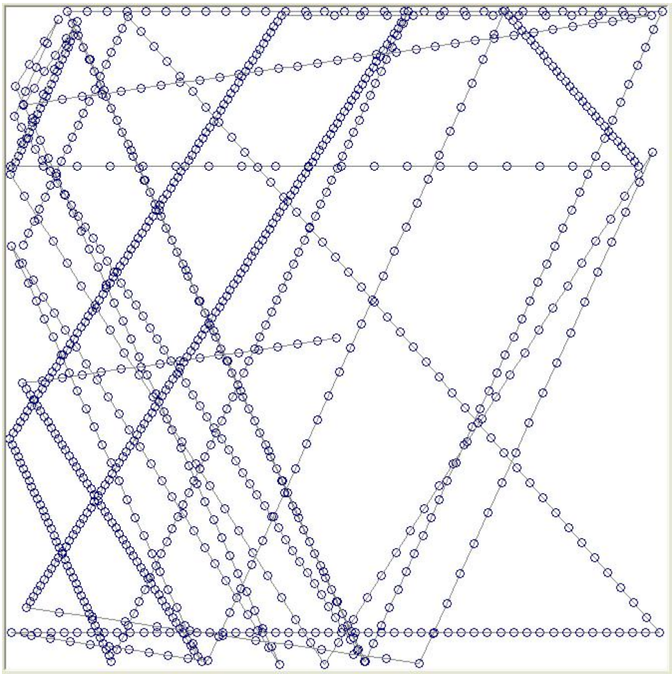


Figure 2.4: Random Direction mobility Model

Figure 2.4 shows an example path of an MN, which begins in the center of the simulation area or position (100, 100), using the Random Direction Mobility Model. The dots in the figure illustrate when the MN has reached a border, paused, and then chosen a new direction. Since the MNs travel to, and usually pause at the border of the simulation area, the average hop count for data packets using the Random Direction Mobility Model will be much higher than the average hop count of most other mobility models (e.g., Random Waypoint Mobility Model). In addition, network partitions will be more likely with the Random Direction Mobility Model compared to other mobility models.

2.5 Conclusion

Traces provide perfect simulation environment to study the performance of different protocols. But feasibility and availability of traces is the main constraint, Synthetic mobility models is an alternative solution. Synthetic mobility models help in realistically model the Ad Hoc wireless networks as the nodes in these types of networks continue to move and the network routers and communication links are fickle. These mobility models help in assessing the Ad Hoc network algorithms in different situations.

Chapter 3

Routing in Ad Hoc Networks

3.1 Introduction

The key task of communication networks whether it may be a wired network or an ad hoc wireless network is to provide end to end connectivity. Easiest way to deliver a packet is to flood a packet in the network. This being a simple method it has many side effects as shown in literature[9]. Then the alternative methods are to identify routes apriori to the packet transmission and transfer the packets over the routes. As the state (connectivity, capacity, offered load) of network components changes often in ways that cannot be predicted the routing algorithm must be intelligent enough to cope up with the changes.

Ideally the control functions that govern the network's performance should meet two objectives [10]:

1. To respond **rapidly** and **correctly** when adapting the network's behavior to current network state.

2. To *minimize* consumption of networks transmission, processing and storage **resources** both during the adoption process and as a result of its adoption.

These objectives, however, are competing, not complementary, and hence much of the work on network control has focused on the tradeoff between *accuracy and efficiency*

The performance of the control structures depend on the underlying network structure. The structure of the wired networks is almost stable so the identified routes work for quite a long period. Here the majority of route failures are due to link failure or node failure. Mobility has no effect on these networks. But, in the context of Ad Hoc networks, nodes move rapidly, the links break and new links form instantaneously; hence the identified routes become stale quickly. The extremely performing algorithm for wired network fails for ad hoc networks.

3.2 Classification of routing algorithms

Since the advent of DARPA packet radio networks in the early 1970s [11], numerous protocols have been developed for ad-hoc mobile networks. Such protocols must deal with the typical limitations of these networks, which include high power consumption, low bandwidth, and high error rates. As shown in Figure 3.1, these routing protocols may generally be categorized as:

- table-driven (Proactive protocols)
- source-initiated on-demand driven (Reactive protocols)
- Hybrid Routing protocols

Despite being designed for the same type of underlying network, the characteristics of each of these protocols are quite distinct.

3.3 Proactive algorithms

Proactive Algorithms(protocols) maintain unicast routes between all pairs of nodes regardless of whether all routes are actually used. Therefore, when the need arises (i.e., when a traffic source begins a session with a remote destination), the traffic source has a route readily available and does not have to incur any delay for route discovery. These protocols also find optimal routes (shortest paths) given a model of link costs.

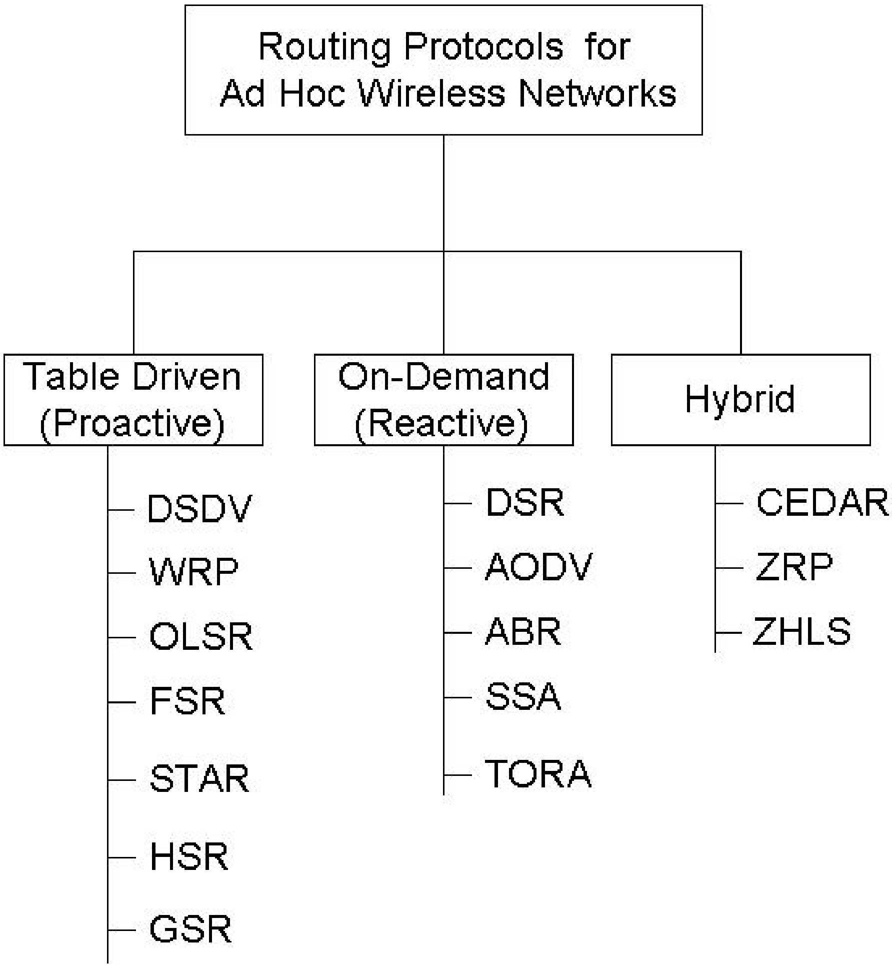


Figure 3.1: Classification of Routing Protocols

Routing protocols on the Internet (i.e, distance vector-based RIP [29] and link state-based OSPF [28]) fall under this category. However, these protocols are not directly suitable for resource-poor and mobile ad hoc networks because of their high overheads and/or somewhat poor convergence behavior. Therefore, several optimized variations of these protocols have been proposed for use in ad hoc networks. In the rest of this section, protocols from this category that have received wide attention are described.

3.3.1 Destination-Sequenced Distance-Vector Routing (DSDV)

The Destination-Sequenced Distance-Vector Routing protocol (DSDV) described in [30] is based on the classical Bellman-Ford routing mechanism [31]. The improvements made to the Bellman-Ford algorithm include freedom from loops in routing tables.

Every mobile node in the network maintains a routing table in which all of the possible destinations within the network and the number of hops to each destination are recorded. Each entry is marked with a sequence number assigned by the destination node. The sequence numbers enable the mobile nodes to distinguish stale routes from new ones, thereby avoiding the formation of routing loops. The node looks up for destination in the routing table and forwards using next hop information given the routing table.

Routing table updates are periodically transmitted throughout the network in order to maintain table consistency. These updates are done as infrequently as possible i.e., when the link state is changed or a node state is changed. As the nodes in the network are mobile the links break very frequently, triggers a lot of updates and clogs the entire the entire network with update packets even at lower mobility.

3.3.2 Wireless Routing Protocol (WRP)

The algorithms of this class use the next hop and second-to-last hop information to overcome the counting-to-infinity problem. this information is sufficient to locally determine the shortest path spanning tree at each node. In these algorithms, every node is updated with the shortest path spanning tree of each of its neighbors. Each node uses the cost of its adjacent links along with shortest path trees reported by neighbors to update its own shortest path tree; the node reports changes to its own shortest path tree to all the neighbors in the form of updates containing distance and second-to-last hop information to each destination.

WRP [32] improves on the earlier algorithms by verifying the consistency of second-to-last hop reported by all neighbors. With this mechanism, WRP reduces the possibility of temporary routing loops, which in turn results in faster convergence time. One major drawback of WRP is its requirement for reliable and ordered delivery of routing messages.

3.3.3 Optimized Link State Routing

OLSR[46] uses the concept of Multipoint Relays (MPRs) to efficiently disseminate link state updates across the network. Only the nodes selected as MPRs by some node are allowed to generate link state updates. Moreover, link state updates contain only the links between MPR nodes and their MPR-Selectors in order to keep the update size small. Thus, only partial topology information is made available at each node. However, this information is sufficient for each to locally compute shortest hop path to every other node because at least one such path consists of only MPR nodes.

OLSR uses only periodic updates for link state dissemination. Since the total overhead is then determined by the product of number of nodes generating the updates, number of nodes forwarding each update and the size of each update, OLSR reduces the overhead compared to a base link state protocol when the network is dense. For a sparse network, OLSR degenerates to traditional link state protocol. Finally, using only periodic updates makes the choice of update interval critical in reacting to topology changes.

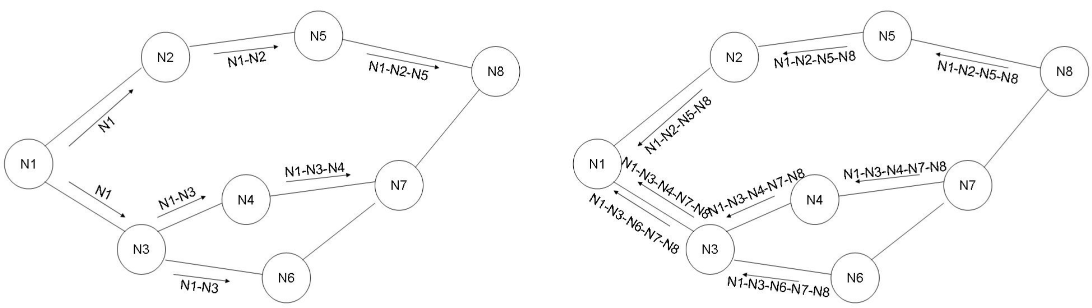
3.4 Reactive algorithms

On-demand (reactive) routing presents an interesting and significant departure from the traditional proactive approach. Main idea in on-demand routing is to find and maintain only needed routes. We will recall that proactive routing protocols maintain all routes without regard to their ultimate use. The obvious advantage with discovering routes on-demand is to avoid incurring the cost of maintaining routes that are not used. This approach is attractive when the network traffic is sporadic, bursty and directed mostly toward a small subset of nodes. However, since routes are created when the need arises, data packets experience queuing delays at the source while the route is being found at session initiation and when route is being repaired later on after a failure. Another, not so obvious consequence of on-demand routing is that routes may become suboptimal, as time progresses since with a pure on-demand protocol a route is used until it fails. In the rest of this Section, three well-known on-demand protocols are discussed.

3.4.1 Dynamic Source Routing

DSR [26] [33] is characterized by the use of source routing. That is, the sender knows the complete hop-by-hop route to the destination. These routes are stored in a route cache. The data packets carry the source route in the packet header.

When a node in the ad hoc network attempts to send a data packet to a destination for which it does not already know the route, it uses a route discovery process to dynamically determine such a route. Route discovery works by flooding the network with route request (also called query) packets as shown in Figure 3.3(a). Each node receiving a request, rebroadcasts it, unless it is the destination or it has a route to the destination in its route cache. Such a node replies to the request with a route reply packet that is routed back to the original source. Route request and reply packets are also source routed. The request builds up the path traversed so far. The reply routes back to the source by traversing this path backward. The route carried back by the reply packet is cached at the source for future use. If any link on a source route is broken (detected by the failure of an attempted data transmission over a link, for example), a route error packet is generated. Route error is sent back toward the source which erases all entries in the route caches along the path that contains the broken link. A new route discovery must be initiated by the source, if this route is still needed and no alternate route is found in the cache.



(a) Route Request (b) Route Reply

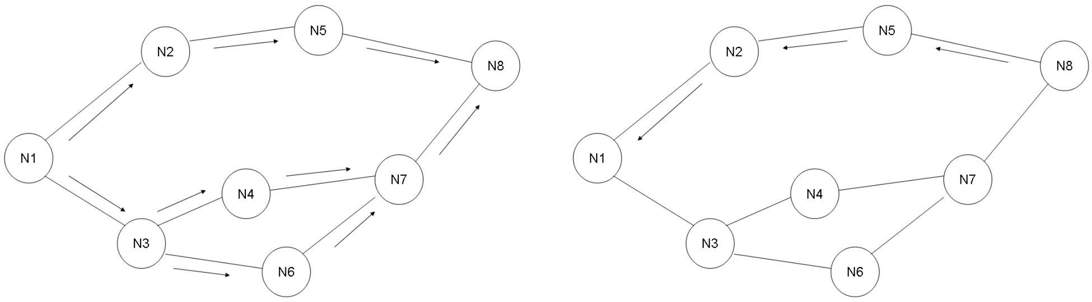
Figure 3.2: Dynamic Source Routing

DSR makes aggressive use of source routing and route caching. With source routing, complete path information is available and routing loops can be easily detected and eliminated without requiring any special mechanism. Because route requests and replies are both source routed, the source and destination, in addition to learning routes to each other, can also learn and cache routes to all intermediate nodes. Also, any forwarding node caches any source route in a packet it forwards for possible future use. DSR employs several optimizations including promiscuous listening which allows nodes that are not participating in forwarding to overhear on-going data transmissions nearby to learn different routes free of cost. To take full advantage of route caching, DSR replies to all requests reaching a destination from a single request cycle. Thus the source learns many alternate routes to the destination, which will be useful in the case that the primary (shortest) route fails. Having access to many alternate routes saves route discovery floods, which is often a performance bottleneck. This may, however, result in route reply flood unless care is taken.

However, aggressive use of route caching comes with a penalty. Basic DSR protocol lacks effective mechanisms to purge stale routes. Use of stale routes not only wastes precious network bandwidth for packets that are eventually dropped, but also causes cache pollution at other nodes when they forward overhear stale routes.

3.4.2 Ad hoc On-demand Distance Vector

AODV [7] [40] shares DSR's on-demand characteristics in that it also discovers routes on an "as needed" basis via a similar route discovery process. However, AODV adopts a very different mechanism to maintain routing information. It uses traditional routing tables, one entry per destination. This is in contrast to DSR, which can maintain multiple route cache entries for each destination. Without source routing, AODV relies on routing table entries to propagate a RREP back to the source and, subsequently, to route data packets to the destination. AODV uses destination sequence numbers as in DSDV [30] to prevent routing loops and to determine freshness of routing information. These sequence numbers are carried by all routing packets.



(a) Route Request (b) Route Reply

Figure 3.3: Ad Hoc On-Demand Distance vector Routing

The absence of source routing and promiscuous listening allows AODV to gather only a very limited amount of routing information with each route discovery. Besides, AODV is conservative in dealing with stale routes. It uses the sequence numbers to infer the freshness of routing information and nodes maintain only the route information for a destination corresponding to the latest known sequence number; routes with older sequence numbers are discarded even though they may still be valid. AODV also uses a timer-based route expiry mechanism to promptly purge stale routes. Again if a low value is chosen for the timeout, valid routes may be needlessly discarded.

In AODV, each node maintains at most one route per destination and as a result, the destination replies only once to the first arriving request during a route discovery. Being a single path protocol, it has to invoke a new route discovery whenever the only path from the source to the destination fails. When topology changes frequently, route discovery needs to be initiated often which can be very inefficient since route discovery flood is associated with significant latency and overhead.

3.4.3 Temporally Ordered Routing Algorithm

TORA [39] is another on-demand protocol. TORA's route discovery procedure computes multiple loop-free routes to the destination which constitute a destination-oriented directed acyclic graph (DAG).

While the ad hoc network is looked upon as an undirected graph, TORA imposes a logical directionality on the links. TORA employs a route maintenance procedure requiring strong inter-nodal coordination based on a link reversal concept proposed in a seminal work by Gafni and Bertsekas [37] for localized recovery from route failures. The basic idea behind link reversal algorithms is as follows. Whenever a link failure at a node causes the node to lose all downstream links to reach the destination (and thus no longer in a destination oriented state), a series of link reversals starting at that node can revert the DAG back to a destination-oriented state.

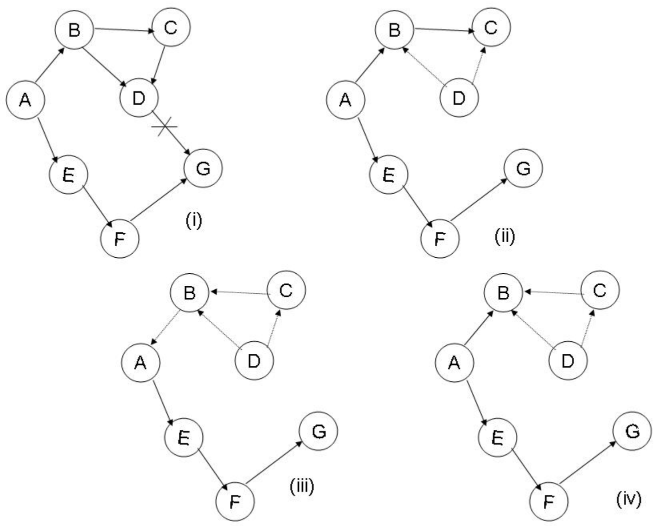


Figure 3.4: Temporally Ordered Routing Algorithm

There are two types of link reversal algorithms namely full reversal and partial reversal differing in the way links incident on a node reverse their direction during the link reversal process. TORA specifically uses a modified version of partial link reversal technique. This modified version allows TORA to detect network partitions as shown in Figure 3.4, a useful feature absent in many ad hoc networking protocols.

3.5 Observation

As research on routing for ad hoc networks have matured, the superiority of one approach over the other has been debated. This question has motivated several simulation-based performance comparison studies [35] [36] [38] [34] and some theoretical studies (e.g., [41]). No clear winner emerged, although on-demand approach usually provides better or similar efficiency relatively for most common scenarios.

All routing algorithms except DSDV rely on flooding for route identification. All algorithms perform fairly well when network is static or quasi static. At the higher mobility the *link_break_count* significantly increases and number of route updates increases. During flooding each intermediate node rebroadcasts a non-forwarded packet. This may cause contention because of reception of the message at neighboring nodes simultaneously. This can be avoided by implementing random back off timers. But the number of retransmissions needed to deliver a message at the destination is significantly less when compared to the message actually being retransmitted. The overhead in doing this can significantly be reduced by using minimum sized forwarding set, construction of which is explained in the next two chapters.

Chapter 4

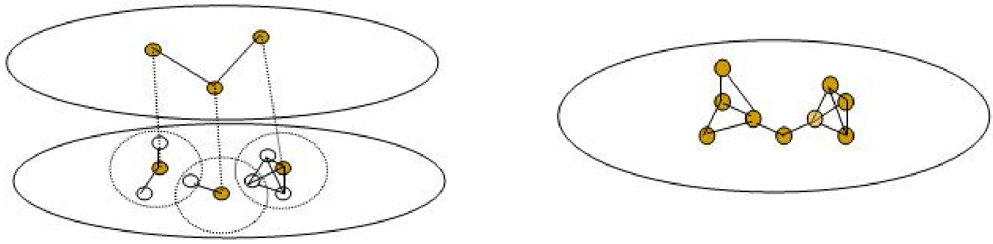
Clustering

4.1 Introduction

Obtaining a hierarchical organization of network is a well-known and studied problem of distributed computing. It has been proven effective in the solution of several problems, such as, minimizing the amount of storage for communication information (e.g., routing and multicast tables), thus reducing the information update overhead, optimizing the use of the network bandwidth, distributing resources throughout the network, etc.

Partitioning of nodes into logical groups (clusters) is called clustering. In the context of ad hoc wireless networks as the nodes are mobile the nodes in the groups get disconnected and then the cluster structure has to be reorganized. In addition, clustering is crucial for controlling the spatial reuse of shared channel (e.g., in terms of time division or frequency division schemes), for minimizing the amount of data to be transmitted in order to maintain routing and control information in a mobile environment, as well as for building and maintaining cluster based virtual network architectures.

Routing algorithms like WRP, OSLR, DSR, TORA, AODV depend upon flooding for route identification. As the density of nodes in the network increases, the inter node contention for broadcast message retransmission increases. This will cause in lost messages and eventual drop of messages which will cause a failure in route identification. The hierarchical organization of network reduces the size of forwarding set. This will lead to



(a) Hierarchical organization (b) Flat Organization

Figure 4.1: Hierarchical vs Flat Organization

a drastic improvement in the network bandwidth utilization. The size of forward set is n and the number of retransmissions of broadcast message is O(n).

The hierarchical organization of network can eventually be done by using multi cluster architecture. Figure 4.1(a). shows hierarchical organization of network, achieved with multi-cluster architecture with duel power mode for inter-cluster communication, while Figure 4.1(b) shows the flat architecture. Nodes in the top oval of Figure 4.1(a). are forwarding nodes while all the nodes in Figure 4.1(b) are forwarding nodes.

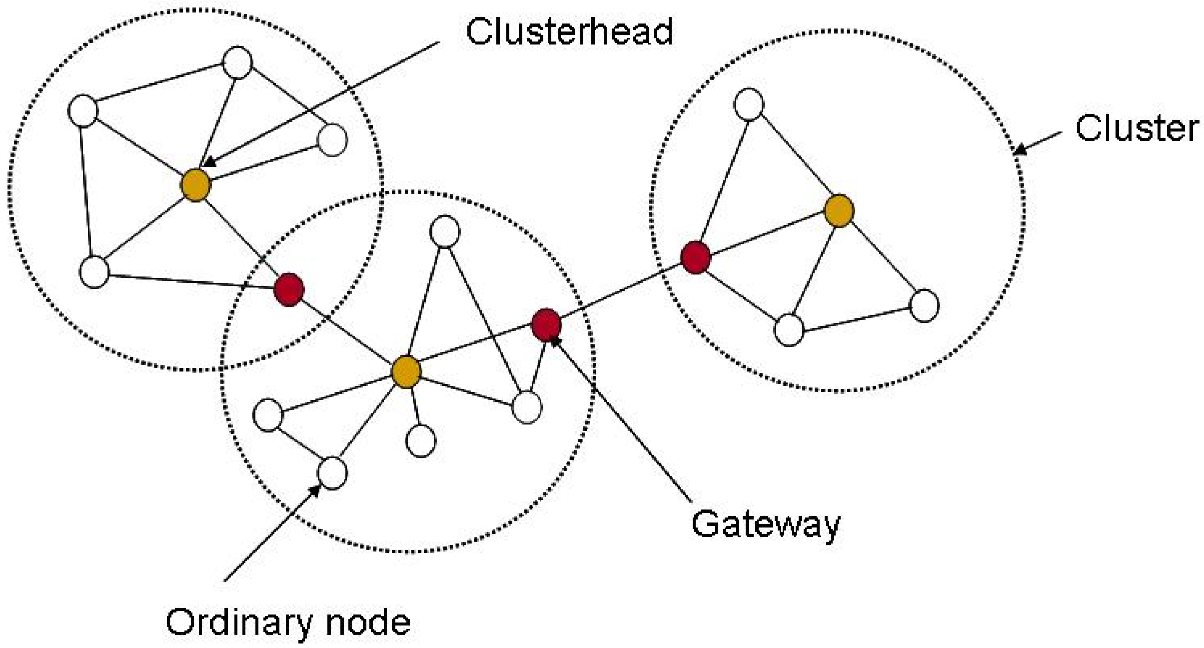


Figure 4.2: Cluster Head Gateway Routing

Hierarchical organization of network gives better location management and improves the scalability of networks. While the flat architecture gives optimal routes and have low power consumption while not being scalable. But the routes formed are suboptimal and leads to single points of failure. The effect being little when compared to the advantages obtained with the hierarchical organization.

4.2 Previous Work

In spite of the benefits provided, Clustering of nodes comes with overhead of formation of clusters and maintenance. An ideal clustering algorithm should have little communication and computation requirements, the clusters formed by the algorithm should be stable and the clustering operation is to be distributed.

Many clustering algorithms [4], [5], [6], [13], [14], [42], [43], [15] are defined in literature. Thesis presents study of some of the popular clustering algorithms which include highest degree, lowest id, node weight based clustering algorithms.

4.2.1 Highest-Degree heuristic

The Highest-Degree, also known as connectivity-based clustering, was originally proposed by Gerla and Parekh [13, 14]. Each node broadcasts its id to the nodes that are within its transmission range. A node x is considered to be a neighbor of another node y if x lies within the transmission range of y. The node with maximum number of neighbors (i.e., maximum degree) is chosen as a clusterhead and any tie is broken by the unique node id's. The neighbors of a clusterhead become members of that cluster and can no longer participate in the election process. Since no clusterheads are directly linked, only one clusterhead is allowed per cluster. Any two nodes in a cluster are at most two-hops away

The size of forward set is n and the number of retransmissions of the message is O(n).As the number of nodes in a cluster is increased, the throughput drops and hence a gradual degradation in the system performance is observed. The reaffiliation count of nodes is high due to node movements and as a result, the highest-degree node (the current clusterhead) may not be re-elected to be a clusterhead even if it looses one neighbor. All these drawbacks occur because this approach does not have any restriction on the upper bound on the number of nodes in a cluster.

4.2.2 Lowest id

The lowest-ID, also as known as identifier based clustering, was originally proposed by Baker and Ephremides [21]. This heuristic assigns a unique id to each node and chooses the node with the minimum id as a clusterhead. Thus, the ids of the neighbors of the clusterhead will be higher than that of the cluster head. However, the clusterhead can delegate its responsibility to the next node with the minimum id in its cluster. A node is called a gateway if it lies within the transmission range of two or more clusterheads. Gateway nodes are generally used for routing between clusters. Only gateway nodes can listen to the different nodes of the overlapping clusters that they lie. The concept of distributed gateway (DG) is also used for inter-cluster communication only when the clusters are not overlapping. DG is a pair of nodes that lies in different clusters but they are within the transmission range of each other. The main advantage of distributed gateway is maintaining connectivity in situations where any clustering algorithm fails to provide connectivity.

The drawback of this heuristic is its bias towards nodes with smaller ids which may lead to the battery drainage of certain nodes. One might think that this problem may be fixed by renumbering the node ids from time to time, which is however non-trivial. There are other problems associated with such renumbering. For instance, the optimal frequency of renumbering would need to be determined so that the system performance is maximized. More importantly, every time node ids are reshuffled, the neighboring list of all the nodes needs also to be changed.

4.2.3 Node-Weight heuristic

These heuristic based algorithms make use of some of the node properties like mobility of nodes, remaining battery power and node degree to calculate their weights. The weights are used to decide upon the methodology for cluster formation. This section describes the clustering algorithms proposed by basagni et al. and das et al.

Distributed Mobility Adaptive Clustering

Basagni et al. [4, 43] proposed two algorithms, namely distributed clustering algorithm and distributed mobility adaptive clustering algorithm (DMAC). In this approach, Each node is assigned weights (a real number) based on its suitability of being a clusterhead. A node is chosen to be a clusterhead if its weight is higher than any of its neighbor's weight; otherwise, it joins a neighboring clusterhead. The smaller node id is chosen in case of a tie. The distributed clustering algorithm makes an assumption that the network topology does not change during the execution of the algorithm. Thus, it is proven to be useful for "quasi-static" networks when the nodes either do not move or move very slowly. The other assumptions are: (i) the messages are guaranteed to be delivered to all of the nodes (neighbors within a finite amount of time), and (ii) every node is aware of the ids and the corresponding weights of all the nodes which are only one hop away. The DMAC algorithm, on the other hand, adapts itself to the network topology changes and therefore can be used for any mobile networks.

To verify the performance of the system, the nodes were assigned weights which varied linearly with their speeds but with negative slope. Results proved that the number of updates required is smaller than the Highest-Degree and Lowest-ID heuristics. Since node weights were varied in each simulation cycle, computing the clusterheads becomes very expensive and there are no optimizations on the system parameters such as throughput and power control.

Weight Based Clustering Algorithm

Das et al has proposed weight based clustering algorithm WCA in [5] and proposed its GA based optimization in [6]. The algorithm utilizes the parameters like node degree, remaining battery power and node mobility to calculate the weights of the nodes and uses. Upon entering the network each node calculates its weight and transmits it weight to the neighbors. The nodes accumulate their weight tables and exchange these tables until every node in the network knows of the weights of the other nodes in the networks. The nodes apply the WCA algorithms to identify the clusters. The maintenance of algorithm is invoke as infrequently as possible i.e., when a node moves out of the reach of all the clusterheads.

The algorithm needs the information of all the nodes in the network to execute, i.e., the size of the network must be known in advance to decide upon when to stop the updating process. This requirement that the nodes need the information of all the nodes in the network causes a no. of rounds of message exchanges. Due to the latency of updating process the convergence time of algorithm increases proportionally with the size of the network. The optimized WCA incorporates a lot of computational over head while producing a small sized dominating set.

4.3 Preliminary Investigation

4.3.1 Problem Definition

An ad hoc wireless network, or simply ad hoc network, can be represented by a unit disk graph [18], where every vertex (host) is associated with a disk centered at this vertex with the same radius (also called transmission range). Two vertices are neighbors (i.e., there is an edge between them) if and only if they are covered by each other's disk.

The network formed by the nodes and the links can be represented by an undirected graph G = (V, E), where V represents the set of nodes and E represents the set of links. Note that the cardinality of V remains the same but the cardinality of E always changes with the creation and deletion of links. Clustering can be thought as a graph partitioning problem with some added constraints. As the underlying graph does not show any regular structure, partitioning the graph optimally (i.e., with minimum number of partitions) with respect to certain parameters becomes an NP-hard problem [44].

4.4 Proposed Weight Based Distributed Clustering Algorithm(DCA)

The proposed clustering algorithm [3] has two phases like cluster setup and cluster maintenance. In the initial cluster setup phase heads are elected by considering the cumulative weights. In the maintenance phase steps are taken to keep the clusters consistent with topology changes. The ideal degree of connectivity, mobility, and remaining battery power are some of the factors whose values are called the weights. Mobility which is denoted as M_v of a node is defined as the average cumulative distance covered by the node up to that moment and is normalized to the scale of [0, 1] where 0 represents maximum mobility where as 1 represents no mobility. A GPS can be a solution to measure the distance covered by a node. When the GPS is not available with the nodes such distance can be measured by calculating the received signal strength [45].

$$M_v = \text{1-(Average displacement/ maximum displacement)}$$

Remaining battery power P_v associated with a node can be calculated using the information like the number of nodes a cluster head is handling and the life time of a cluster head.

$$P_v = \text{remaining battery power / total battery power of mobile nodes.}$$

Degree of a node D_v is the cardinality of a mobile node and is normalized to the scale of [0, 1] where 1 represents that node is associated with an ideal degree and the value decreases as the value is moving out of the ideal degree.

$$D_v = 1 - |\text{Cardinality-Ideal Degree}\ | / \text{ Ideal Degree}$$

The cumulative weight W_v associated with each node is a weighted proportion of parameters associated with each node.

$$W_v = \mathrm{a} * D_v + \mathrm{b} * P_v + \mathrm{c} * M_v$$

Where a, b, c are constants, which decides the proportion of different parameters that will effect the clustering. And the sum of all three parameters a, b, c is assumed to be 1.

During the initial cluster setup phase, nodes in the network compute cumulative weight W_v associated with them and broadcast it. Every node waits for other nodes to transmit their cumulative weights. The nodes acquire the information about its neighbors by these broadcast messages. A node with maximum weight among the neighbors raises the cluster head message and all the nodes in its vicinity affiliate to that cluster head. If a node listens to more than one cluster head message it affiliates with the cluster head to which it has minimum distance. Other nodes that did not receive a cluster head message within a certain waiting period will broadcast cluster head message. The waiting periods associated with each node varies with their weight. Higher the weight associated with a node, lower is its waiting time.

Due to mobility of nodes in the Ad Hoc networks, nodes move from one location to another as a result of which the communication links between nodes are lost and new links are established. The cluster maintenance phase considers the topology changes and reacts as follows. When a new link is formed, A Single node which is not in the vicinity of any other cluster head elects itself to be a cluster head. When such a cluster head comes in contact with another cluster head having a higher weight than it, then it affiliates itself to that cluster head. There occurs a Link Failure when a member node moves out of the vicinity of its existing cluster head, then it affiliates itself with a new cluster head in its close vicinity. When a node moves out of the vicinity of all the cluster heads it forms a single node cluster.

Cluster Head Reorganization: When a cluster has a node with a weight more than the cluster head by as its member then the existing cluster head gives up the cluster head token to such a node, all the member nodes will affiliate to the newly elected cluster head. can be any value between [0, 1] but it gives better results when its values are around 0.2. When any cluster head drains out of network resources a global reelection of cluster heads needs to take place.

4.5 Performance Evaluation

Lowest id, Highest Degree, DMAC and DCA comes under distributed algorithms have the same complexity for algorithm execution. The WCA, a centralized has different algorithm execution complexity. The thesis reviews the communication overhead associated with the clustering setup and maintenance phases of the DMAC, WCA and DCA algorithms.

The WCA requires the information of every node to be present at every other node in the network to start the algorithm execution. That is every node requires the link state of all the nodes in the network and the weights of the nodes. This is done by the process of aggregation and rebroadcast. Every node broadcasts its local state. A node upon receiving the local state update packet updates its update packet and retransmits it. The process continues until no new broadcasts are heard. The nodes execute the algorithm to find cluster heads and the nodes which are affiliated to these clusterheads. The algorithm requires $O(NLog(N))$ messages to get the complete network topology, where, N is the no. of nodes in the network. and computational complexity is $O(N^2)$.

DCA and DMAC are election based algorithms. A node contends to become a cluster head, nodes with higher weight contends to become a cluster head. the total no. of messages that are transmitted in the drew are $2 * N_i$ as shown by Basagni et al. so the communication overhead associated with the DMAC is $O(N_i)$. The Computational complexity of the algorithm is $O(N_i)$.

The communication and computation complexities shown above for DMAC and DCA are very attractive but the rate of activation of algorithm for DMAC and DCA are relatively high. WCA while providing the optimal sized clusters has very high communication overhead.

4.6 Simulation Study

The simulation of Ad Hoc network with N nodes is done in an area of 100 X 100 twips assuming all the nodes are similar in capabilities like Transmission range available Memory and the Battery power associated with the nodes. The nodes in the network can move a distance of [0, MaxDist] in any direction in a simulation clock tick.

To measure the performance of propsed algorithm DCA, three parameters are identified:

- Reaffiliation Count
- Dominating Set Size
- Dominating Set Updates

At any point of time cardinality of dominant set gives the number of clusterheads or the dominating set size The reaffiliation count is incremented when a node gets dissociated from its clusterhead and becomes a member of another cluster within the current dominant set. The dominant set update takes place when existing clusterhead set changes. These three parameters are studied for varying number of nodes (N) in the system and transmission range.

In simulation experiments conducted, N was varied between 20 and 60, and the transmission range was varied between 0 and 70. The nodes moved randomly in any possible direction θ with a maximum displacement of 10 where, θ is varied between $[0, 2\pi]$. Assume that each clusterhead can handle at most d = 10 nodes (ideal degree) in its cluster in terms of resource allocation. Due to the importance of keeping the node degree as close to the ideal as possible, the weight a associated with D_v was chosen high. Mobility and battery power were given low weights. The values used for simulation were a = 0.8, b = 0.1 and c = 0.1. Note that these values are arbitrary at this time and should be adjusted according to the system requirements.

Dominating Set Update causes the existing routes to be recomputed as existing cluster heads resigns and the new cluster heads are elected, routes has to pass through new cluster heads. To minimize *route recomputation* the dominating set updates has to be kept minimum. Figure 4.3 shows the number of dominant set updates with respect to the transmission range.

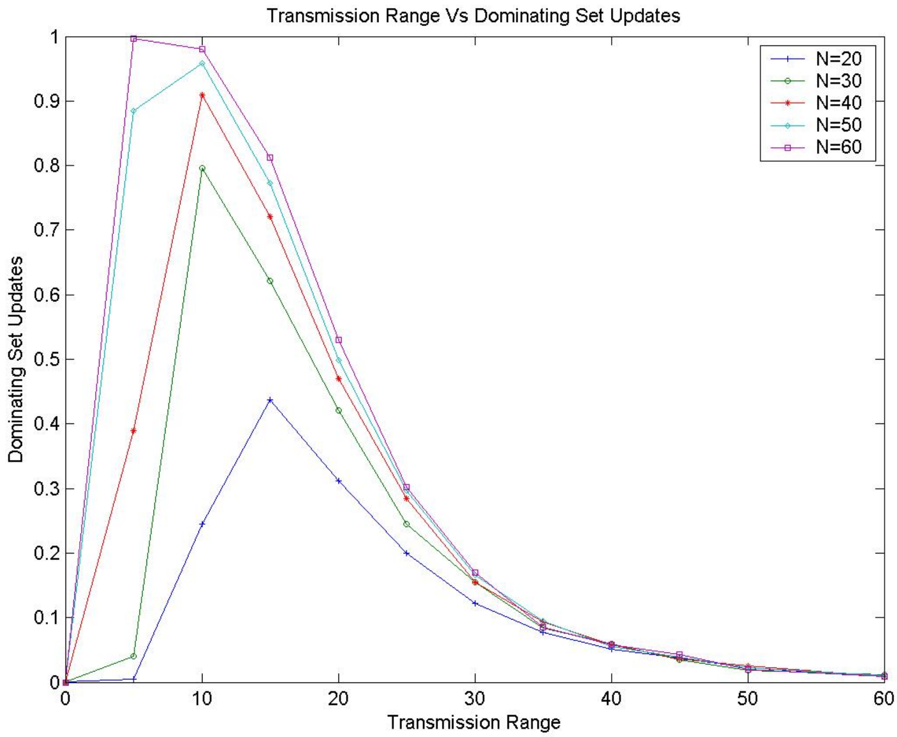


Figure 4.3: Graph Showing Dominating Set updates

For smaller transmission range, the cluster area is small and the probability of a node moving out of its cluster is high. As the transmission range increases, the number of dominant set updates decreases because the nodes stay within their cluster in spite of their movements.

Reaffiliations causes the routes initiated and passing through the node be recomputed, so keeping the reaffiliations as low as possible is the goal of any clustering algorithm. Figure 4.4 shows the reaffiliations per unit time. For low transmission range, the nodes in a cluster are relatively close to the clusterhead, and a detachment is unlikely.

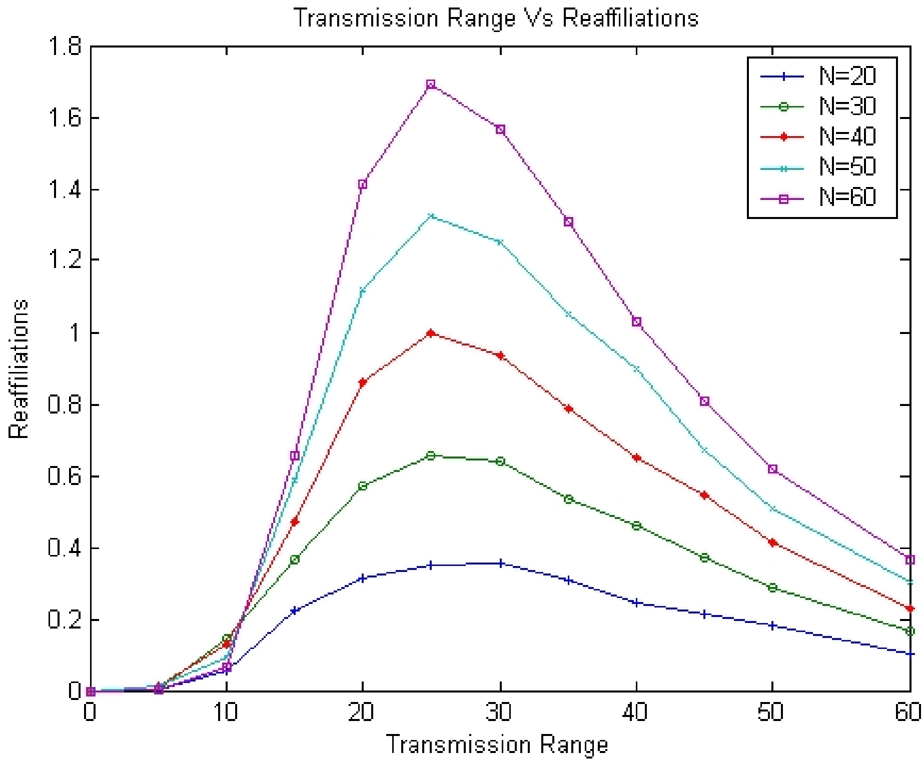


Figure 4.4: Graph Showing Reaffiliation Count

The number of reaffiliations increases as the transmission range increases, and reaches a peak when transmission range is between 25 and 30. Further increase in the transmission range results in a decrease in the reaffiliations since the nodes, in spite of their random motion, tend to stay inside the large area covered by the clusterhead.

Figure 4.5 shows the variation of the average number of clusterheads with respect to the transmission range where MaxDisp of 5. The results are shown for varying N. The average number of clusterheads decreases with the increase in the transmission range. This is due to the fact that a clusterhead with a large transmission range will cover a larger area.

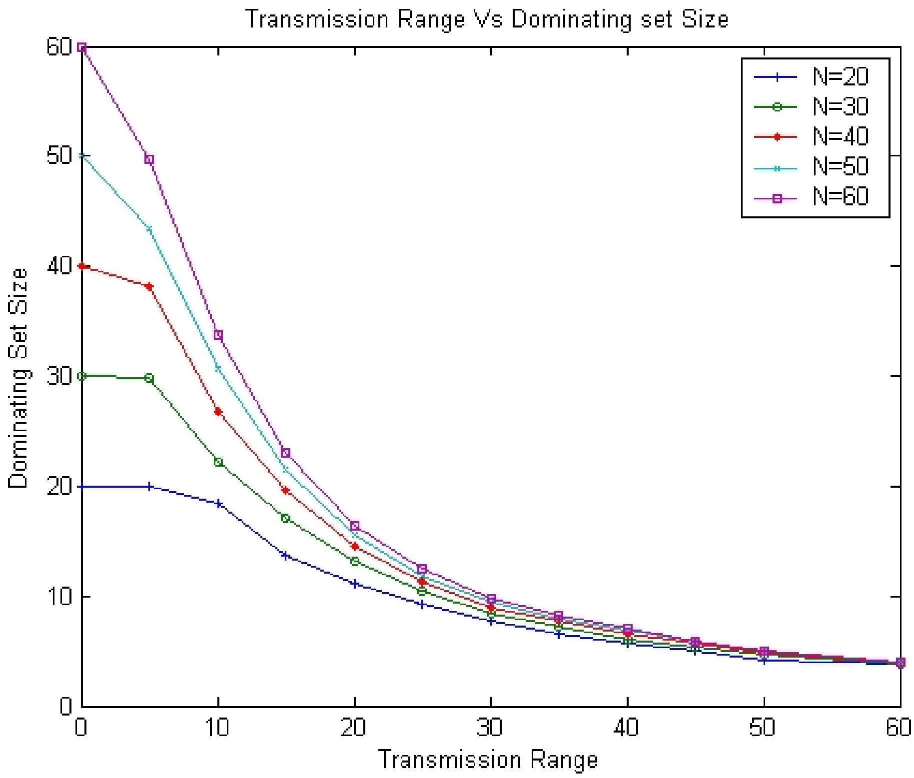


Figure 4.5: Graph Showing Dominating Set Sizes

Figure 4.6 shows the comparison of the reaffiliation count of lowest id, DMAC, WCA and proposed DCA with N = 30 and a MaxDisp = 5 simulation results show that proposed DCA performs in comparison to that of DMAC and WCA and out performs Lowest id.

4.7 Conclusion

This thesis proposed a Weight Based Distributed Clustering Algorithm which can dynamically adapt itself to the ever changing topology of Ad Hoc wireless networks. The cluster heads are selected locally based on the weights computed. The algorithm tries to form

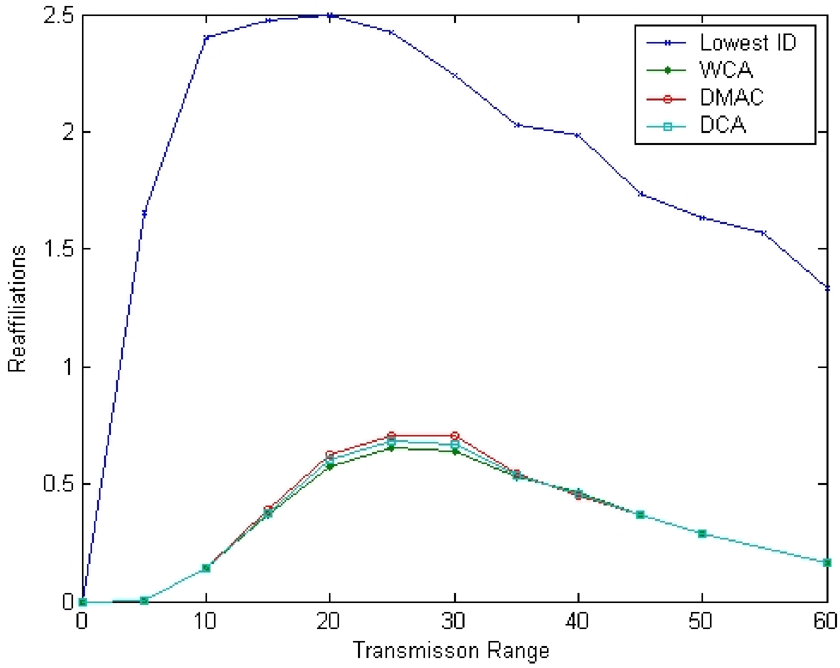


Figure 4.6: Graph Showing comparison of Reaffiliations

optimal sized clusters. DCA considers the parameters of the mobile devices like mobility, remaining battery power and ideal degree in cluster formation. The algorithm can be tuned to the requirements by adjusting different parameters. The algorithm is expected to perform marginally better than the existing algorithms but it describes clearly the normalization of weights and has given more emphasis on cluster reorganization.

Chapter 5

Connected Dominating sets

5.1 Introduction

Routing protocol design is one of the challenging issues in ad hoc networks. The routing information in the routing tables becomes stale very frequently in ad hoc wireless networks as compared to that of wired counterpart. The routing algorithms mostly depend on broadcasting for route identification. The optimization of broadcast step in ad hoc wireless networks improves the performance and scalability a lot. The cluster based routing can be done in two ways. Either by using gateway nodes or by using duel power mode. Gateway identification requires three hop neighborhood information while duel power mode consumes extra battery power.

In packet cellular networks, taking advantage of the backbone, routing algorithms can support flow, Multicasting and even fault-tolerant routing for mobile computers. This thesis impose a virtual backbone structure on the ad-hoc network, in order to support unicast, Multicast, and fault-tolerant routing within the ad-hoc network.

This virtual backbone differs from the wired backbone of cellular networks in two key ways: (a) it may change as nodes move, and (b) it is not used primarily for routing packets or flows, but only for computing and updating routes. The primary routes for packets and flows are still computed by a shortest-paths computation. the virtual backbone can, if necessary, provide backup routes to handle interim failures. Because of the dynamic nature of the virtual backbone, CDS approach splits the routing problem into two levels: (a) find and update the virtual backbone, and (b) then find and update routes.

To keep the virtual backbone as small as possible, an approximation to the minimum connected dominating set is identified.

5.2 Related Work

Many people have tried to obtain the approximation to Minimum Connected Dominating Sets, it includes [1], [2], [8], [16], [17], [19], [20], [12] and [27]. Algorithms that construct a CDS in ad hoc networks can be divided into two categories: centralized algorithms that depend on network-wide information for coordination and decentralized that depend on local information only. The centralized algorithms provide a minimum sized connected dominating set while utilizing the global state information while others utilize the local information.

The algorithms that form connected dominating sets include but not limited to MCDS by S. Guha and S. Khuller [12], Rule 1 & Rule 2 proposed by Jie Wu [2], rule k proposed by Fei Dai and Jie Wu [1] and Stojmenovic's algorithm [8]. It is shown in literature that the performance of Stojminovic's algorithm is similar to that of the rule k proposed by the Fei Dai and Jie Wu. MCDS is a global algorithm while other algorithms are localized.

5.2.1 Minimum Connected Dominating Sets

Minimum Connected Dominating Sets(MCDS), explained in "Approximation Algorithms for Connected Dominating Sets" by S.Guha and S.Khuller [12], is a centralized algorithm to find connected dominating sets. The algorithm takes as input a strongly connected graph and gives an approximation to the connected dominating sets as an output. The algorithm runs in two phases.

At the start of the first phase all nodes are colored white. Each vertex included in the dominating set is colored black. Nodes that are dominated are colored gray (once they are adjacent to a black node). In the first phase the algorithm picks a node at each step and colors it black, coloring all adjacent white nodes gray. A piece is defined as a white node or a black connected component. At each step, pick a node to color black that gives the maximum (nonzero) reduction in the number of pieces. At the end of this phase if no vertex

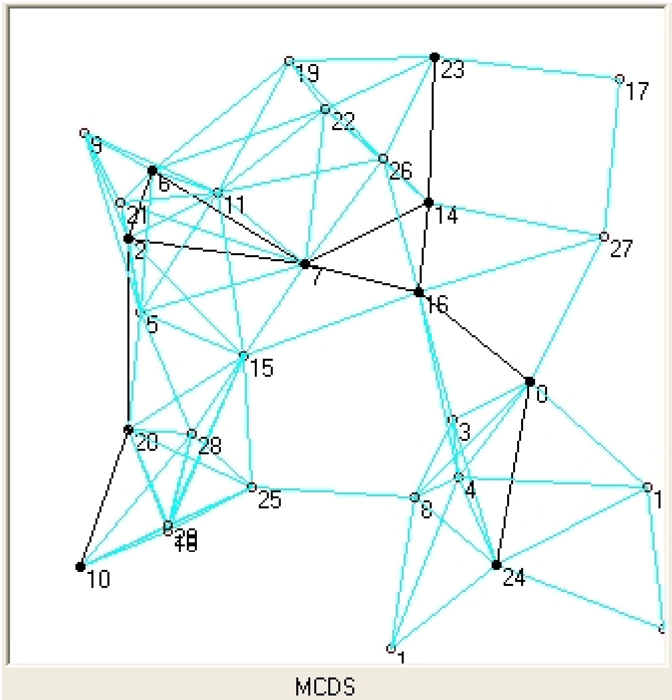


Figure 5.1: Minimum Connected Dominating Set

gives a nonzero reduction to the number of pieces, then there are no white nodes left. In the second phase, collection of black connected components are connected to form a single black connected component. Recursively connect pairs of black components by choosing a chain of vertices, until there is one black connected component. The final solution is the set of black vertices that form the connected component. Cds formed by MCDS is shown in figure 5.1. Dark circles in the figure are dominating nodes.

5.2.2 Extended Dominating-Set-Based Routing

The algorithm is proposed in “Extended Dominating-Set-Based Routing in Ad Hoc Wireless Networks with Unidirectional Links” by Jie Wu [27] and is basically defined for improving performance of broadcasting in networks with unidirectional links. But, broadcast packets for route identification process utilize the backward path traversal.

In the marking process, a vertex is marked T because it may be the only connection between its two neighbors. However, if there are multiple connections available, it is not necessary to keep all of them. A vertex is covered if its neighbors can reach each other via other connected marked vertices.

Extended marking process:

1. Initially assign F as marker to each u belongs to nodes
2. u changes its marker m(u) to T if there exists vertices v and w such that (v,u) and (w,u) belong to edges but (v,w) does not.

Rule 1

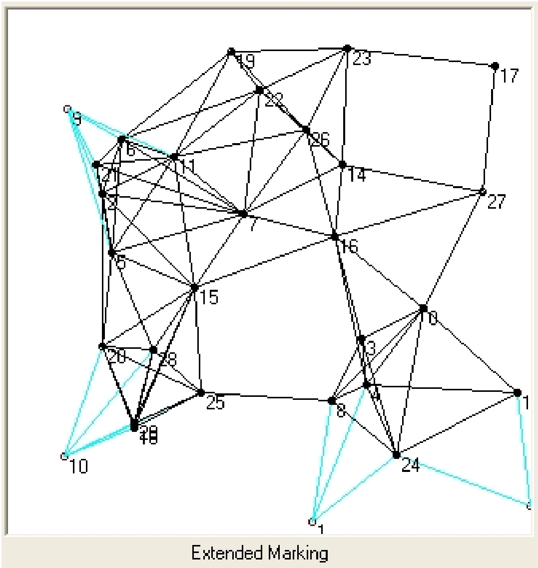
1. Assume that u is a marked vertex and v is a vertex. Unmark u if both conditions hold in D:
2. N(u)-v$\subseteq$ N(v)
3. id(u)$\prec$ id(v)

Rule 2

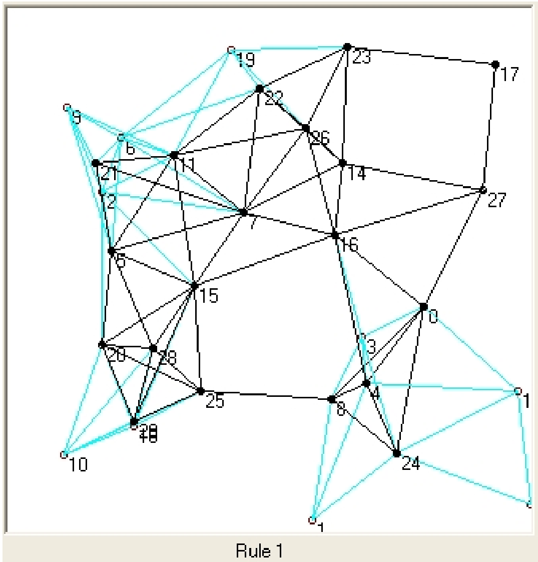
1. Assume that u is a marked vertex and v,w are vertices. Unmark u if both conditions hold in D:
2. N(u)-{ v,w} $\subseteq$N(v)$\cup$ N(u)
3. id(u)is min{id(u),id(v),id(v)}

Algorithm: Proposed by Jie Wu

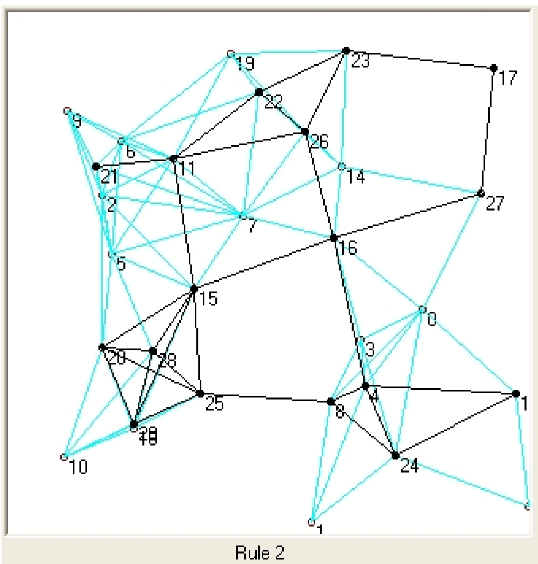
Two dominant pruning rules proposed to reduce the size of the connected dominating set are shown above. The idea is the following: If a vertex is covered by no more than two connected vertices, removing this vertex from dominating set will not compromise its functionality as a CDS. To avoid simultaneous removal of two vertices covering each other, a vertex is removed only when it is covered by vertices with higher id's. Node id of each vertex serves as a priority. Nodes with high priorities have high probability of becoming gateways. Id uniqueness is not necessary, but equal id's will produce more gateways.



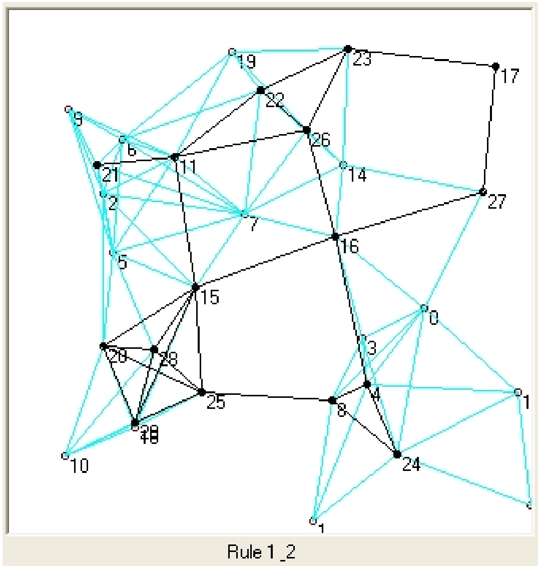
(a) Extended Marking Process



(b) Marking By Rule1



(c) Marking by Rule2



(d) Marking By Rule 1 & Rule 2

Figure 5.2: Jie Wu's Marking process

The node marking my Extended Marking Process is shown in figure 5.2(a). Figure 5.2(b), figure 5.2(c), figure 5.2(d) shows the marking by Rule 1, rule 2 and rule 1 & rule 2 togather. The Dark nodes are members of dominating set while other nodes are nonforwarding nodes.

5.2.3 Generalized Pruning Rule

Fe Dai et al proposed generalized pruning rule given below, which generates an approximation to minimum sized connected dominating set. Nodes are initially marked using Extended Marking Algorithm given above. The algorithm creates strongly connected components with marked neighbors higher in id than the node. If these neighbors form a single strongly connected component and cover other neighbors then the node will be unmarked.

Generalized pruning Rule

Assume that $G' = (V', E')$ is the induced subgraph of a given directed graph $G = (V, E)$ from marked vertex set V'.

Rule K:

Assume that V'_k=$\{v'_1, v'_2, v'_3, ..., v'_k\}$ is the vertex set of a strongly connected subgraph in G'. If $N(u) - V'_k \subseteq N(V'_k)$ where, $N(V'_k) = \cup_{v_i \in V'_k} N(v_k)$ and $id(u) \prec min\{id(1), id(2)....id(3)\}$

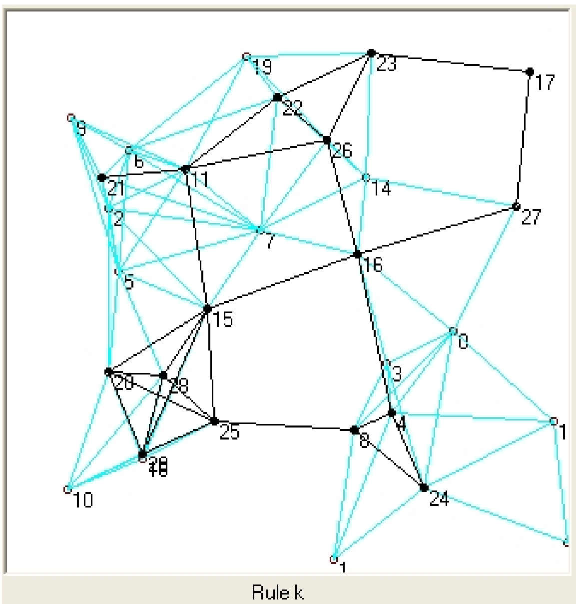


Figure 5.3: Marking by Rule k

The algorithm when executed in unrestricted fashion utilizes the utilizes the information of the entire networks and forms a small connected dominating set but its restricted version is more stable and is less reactive to the global network changes. The restricted version forms 5% large dominating set as compared to its unrestricted version. Figure 5.3 shows restricted implementation of Rule k.

5.3 Preliminary Investigation

The algorithms described in previous section form a connected dominating sets but all these algorithms are reactive to a change in a 2 hop neighborhood or more which can still be reduced to form a more stable CDS. A change in two hop neighborhood activates the algorithm to find out the forwarding status of the node. The increased scope increases the rate of activation of the CDS formation algorithm. The CDS updates doesn't effect existing routes but the marker exchange will cause the extra congestion in the network. The reduction of which will improve stability in highly mobile conditions.

The dependence of the above algorithms on the node ids causes the marking process to be biased towards the higher ids. This will cause a set of nodes to be always marked. While the probability of a lower id node be marked is near zero. This unfair biasness causes a set of network nodes fail soon.

The extended marking process identifies whether the node is in shortest path between any two nodes in its neighborhood, it is not necessary to keep all the nodes that are in shortest path between these two nodes. Selective pruning or unmarking of marked nodes is necessary to minimizing the dominating set size. As shown in the figure below the extended marking process marks all the nodes. To decrease the redundancy a part of these nodes can be pruned from the dominating set such that every node in the network gets a broadcast packet.

The mobility of a node and the remaining battery power associated with the mobile node gives the better approximation of the node status than the unique id associated with the nodes. Considering the node properties will reduce the biasing of the marking process towards a set of nodes. The marking process is more likely towards the nodes with higher remaining battery power means an improvement in the total network life or more biasing towards lower mobility node means more stable CDS.

5.3.1 Problem Definition

An ad hoc wireless network, or simply ad hoc network, can be represented by a *unit disk graph* [18], where every vertex (host) is associated with a disk centered at this vertex with the same radius (also called transmission range). Two vertices are neighbors (i.e., there is an edge between them) if and only if they are covered by each other's disk.

The network formed by the nodes and the links can be represented by an undirected graph G = (V, E), where V represents the set of nodes and E represents the set of links. Note that the cardinality of V remains the same but the cardinality of E always changes with the creation and deletion of links.

5.4 Weight Based Self Pruning Algorithm(SPA)

The network life depends upon how effectively the tasks are distributed between different coordinating members. The algorithm computes the weight associated with each node using the equation 5.1 where, a and b are weights associated with the parameters mobility M_v which is calculated by using equation 5.2 and P_v calculated by equation 5.3 and $a + b$ is assumed to be 1.

$$W_v = a * M_v + b * P_v \tag{5.1}$$

$$M_v = 1 - Avg.Displacement/MaximumDisplacement \tag{5.2}$$

$$P_v = RemainingPower/TotalPower \tag{5.3}$$

Each node in the network makes a local broadcast of information update packet consisting of node weight and the neighbors. Each node updates the neighbor list upon receiving the information update packet. The neighbor list contain list of neighbors along with their weights and list of neighbors.

A node transmits a beckon packet if it did not transmit a packet in that interval. A link is said to be disconnected with a node if no transmission from the node is received in 2 consecutive intervals. Local update packets are piggybacked behind the broadcast packets to reduce the congestion. A wait time proportional to the node weight is introduced to reduce the contention and collision during rebroadcast.

Node u $\in$ V covers v $\in$ V if N(u) $\subseteq$ N(v) where N(u), the neighbor set of u and N(v), the neighbor set of v. A set V_k of nodes covers u if $N(u) - V_k \subseteq N(V_k)$ where, $N(V_k') = \cup_{v_i \in V_k'} N(v_k)$

Node Extended Marking: Assume (V, E) represents a graph

1. Assign F as Marker $M(u)$ to Node u.
2. u changes its Marker M(u) to T if $\exists v, w \in V$ such that $(v, u) \& (w, u) \in E$ but $(v, w) \notin E$

Self Pruning Algorithm

1. If Node Extended Marking(u) is true then do steps 2 to 4 otherwise M(u)=false
2. Check whether the set H, the higher weight neighbors of node u are connected.
3. If H is connected then check whether H will cover the neighbors of node u.
4. If H covers $N(u)$ then assign M(u)=False
5. Return $M(u)$

Algorithm: Weight based Self Pruning Algorithm

The nodes in the network executes the weight based self pruning algorithm given above. The node has a forwarding status if the algorithm returns true otherwise the node simply listens to the broadcast packets. The nodes follow the routing algorithm for making routing decisions like, replying a route request packet. If a non-forwarding node wants to transmit a packet to any other node in the network it simply broadcasts the route request packet. As opposed to the flat architecture the request packet is retransmitted only by the forwarding set, which will increase the efficiency in utilizing **bandwidth**.

The nodes listen to the local update packets to check if there is any change in its neighborhood. Upon change in its neighborhood node computes the SPA algorithm to find its status. The weights of nodes are recomputed and are transmitted in update packets. Nodes show their weights to be 0 if their remaining battery power is below a threshold value. The algorithm forms the connected dominating set as shown in Figure 5.4.

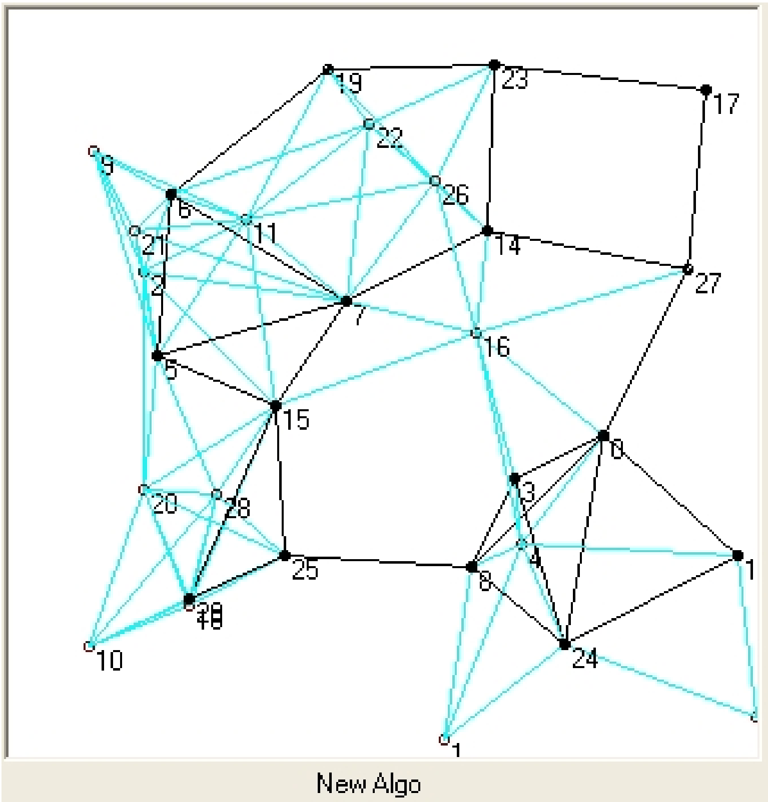


Figure 5.4: CDS Formation by SPA

5.5 Performance Evaluation

The performance of Connected dominating set formation algorithm depends on the message complexity and updation rate of the algorithm. More reactive algorithm maintains fresh data but comes with a draw back of update overhead and computation overhead.

The performance of CDS formation algorithms is analyzed here based on their message complexity and computation complexity.

Rule 1 & Rule 2 and Rule k are considered along with proposed SPA for evaluating performance.Extended marking process is utilized by all these processes, whose computation complexity is $O(N_i^2)$ per node where N_i is the number of neighbors of node i.

Let us assume that nodes are uniformly distributed and each node have equal number of neighbors (N_i). In Rule 1 each node compares its neighbors list with every of the higher order neighbors. If it finds that higher id neighbor marked by Extended Marking process can cover its neighbor hood then it unmarks itself. The computational complexity of rule 1 will be $O(N_i^2)$. Rule 2 compares for every pair of nodes as in the above this gives the computational complexity of $O(N_i^3)$.The effective message complexity will be $O(N_i^3)$ The nodes exchange their markers upon every activation of marking process; this will introduce an additional message complexity $O(N_i)$ messages.

Rule k computes the CDS by finding the strongly connected components, In the restricted implementation of connected dominating sets, this step requires $o(N_i^2)$ computations and the comparison step requires $O(N_i^2)$ computations. Overall the computation complexity of Rule k is $o(N_i^2)$. The algorithm exchanges markers for its execution which, will cause additional message complexity of $O(N_i)$.

DCA performs the operation by calculating the connected components and compares the neighborhood to check whether they covers it. The computation complexity is of $O(N_i^2)$. The algorithm does not require any marker exchange to perform its operations. So the additional message complexity for this operation is 0.

The Rule 1 & Rule 2 algorithms are activated by the change in 3 hop neighbor hood to update their status. where as the Rule K is effected by a change in 2-Hop neighborhood and DCA is effected by changes in 1-hop neighborhood.

5.6 Simulation Study

The simulation of Ad Hoc network with N nodes is done in an area of 100 X 100 twips assuming all the nodes are similar in capabilities like Transmission range available Memory and the Battery power associated with the nodes. The nodes in the network can move a distance of [0, MaxDist] in any direction in a simulation clock tick.

To measure the performance of proposed algorithm SPA, these parameters are identified:

- Dominating Set Size
- Dominating Set distribution

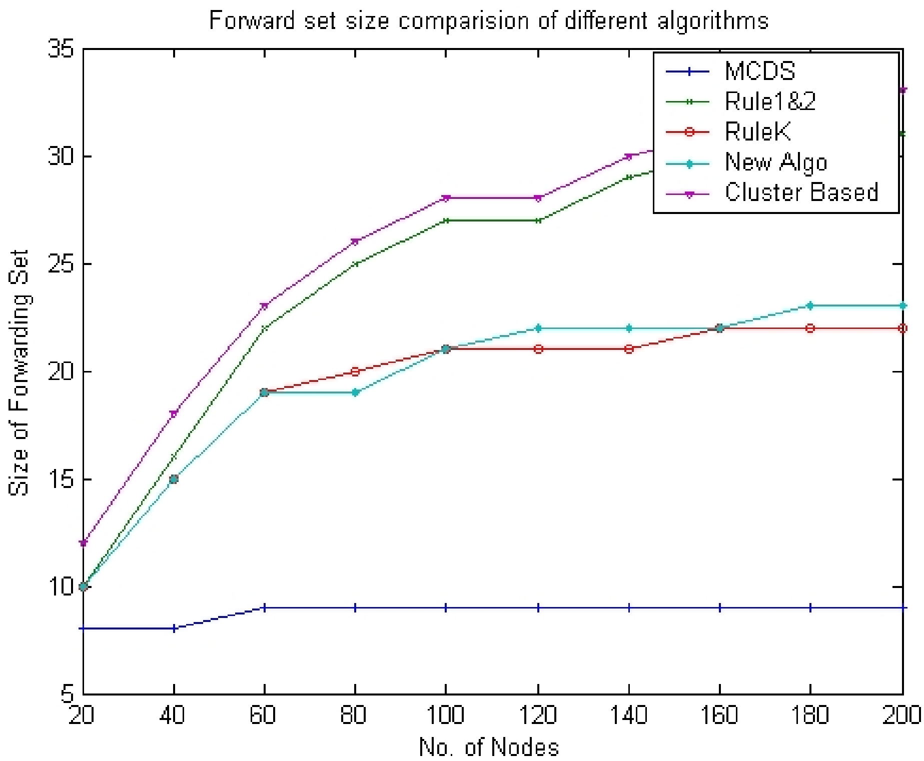


Figure 5.5: Graph: Connected dominating set size

During the process of simulation, N was varied between 20 and 200, and the transmission range was varied between 0 and 70. The nodes moved randomly in any possible direction θ with a maximum displacement of 10 where, θ is varied between $[0, 2\pi]$. Maximum displacement is set at 5.The parameters of weight computation is set at a=0.5 and b=0.5, keeping a note that these values are arbitrary at this time and should be adjusted according to the system requirements.

The dominating set size gives the no. of times a message will be rebroadcasted to distribute it over the entire network. The smaller is the dominating set the lesser is the broadcast overhead associated with the network. The result shown in figure 5.5 show that SPA has better forwarding set when compared to that of Rule 1& Rule 2 and Cluster based approach, but, comparable in performance with Rule k.

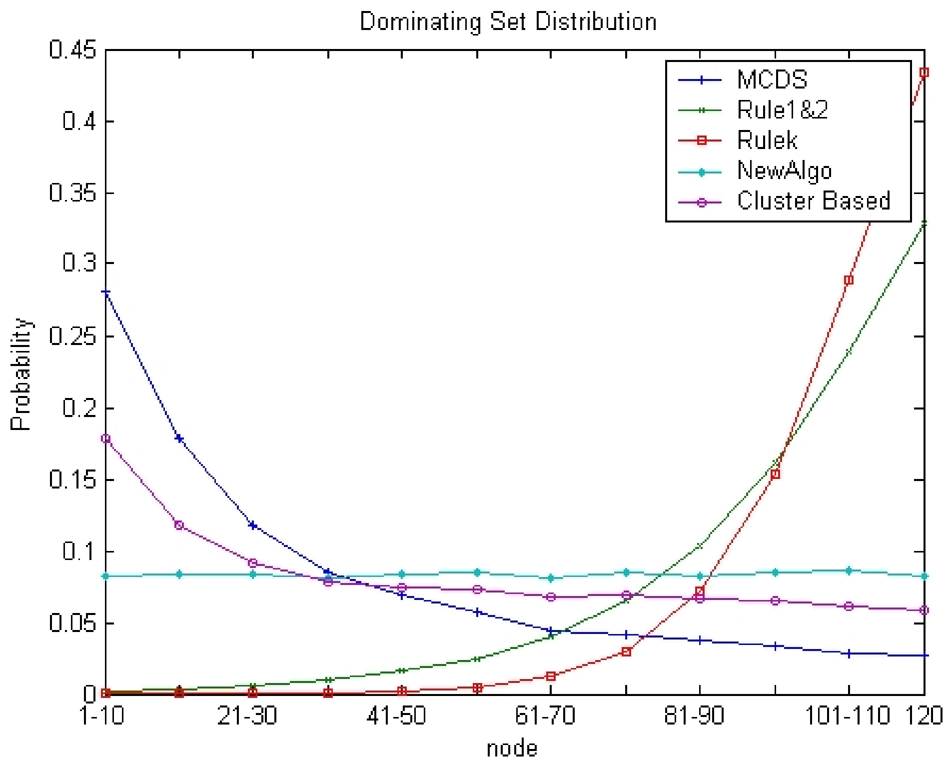


Figure 5.6: Graph: Connected dominating set Distribution

The nodes in the dominating set have the additional task of forwarding the broadcast packets and acting as backup routes for data transmission. Performing additional tasks require additional energy. So, the dominating set formation algorithm should have less biasness towards a set of nodes i.e, towards higher id nodes or lower id nodes. Simulation results show that Rule 1 & Rule 2 and Rule k are biased towards higher order nodes while cluster based approach and MCDS are biased towards lower id nodes. Proposed Weight based Self Pruning Algorithm distributes nodes uniformly over the entire range.

5.7 Conclusion

SPA forms a connected dominating set which works as a broadcast backbone. The algorithm distributed the markers uniformly based on the parameters of the communication devices as against to the biasing by other algorithms towards highest (or lowest) "id" nodes. The algorithm does not have additional communication overhead for its execution as it doesn't exchange any markers. The results show that using connected dominating sets improves the routing while improving broadcast. The flexibility of utilizing any routing algorithm over this virtual backbone helps utilizing the existing routing algorithms without any modifications.

Chapter 6

Conclusion and Future Work

The algorithms that contribute towards achieving scalability and stability of Ad Hoc wireless networks have been proposed in this thesis work. This thesis provided two solutions to the problem of dominating set formation. One provided a cluster based solution while the other one is a self pruning based algorithm. Stability and scalability of Ad Hoc Network can only be achieved by optimally selecting the dominating set. Balancing the load among different heterogeneous nodes having varied communication, computation capabilities and with variable amount of resources while maximizing the network life time is the major task.

In Cluster based approach, a "Weight based distributed clustering algorithm" which dynamically adapts to the ever changing topology of Ad Hoc Wireless Networks have been proposed. The algorithm has the flexibility of assigning different weights to various system parameters (such as Ideal Degree, Maximum Displacement, mobility, battery power, etc.,). The algorithm is executed locally when the current clusters are inconsistent with the topology of the network. This clustering algorithm tries to minimize the communication complexity to the extent of being as low as possible. The proposed piggybacking techniques for information update packets are influential in reducing the communication overhead.

Proposed weight based self pruning algorithm follows the connected dominating set paradigm. The algorithm computes a marker whose value is either true or false. The node forwards a broadcast packet if its marker is true otherwise it simply listens to the same. The mobility and remaining battery power of node have been considered as important factors and the weight of node is taken as weighted proportion of these two parameters. To improve the lifetime of nodes, weight of nodes having remaining battery power below a threshold value is made to be zero. The algorithm has no communication overhead for its execution. The algorithm is not reactive to the changes in global topology but, reactive to the changes in local topology.

The node distribution of SPA, the connected dominating set algorithm, is very encouraging. The algorithm is so flexible that any routing algorithm can be executed without any modification over the formed connected dominating set. The algorithm reduced collision and packet loss during broadcast by introducing variable time gaps for rebroadcast. The changes in the dominating set doesn't affect the existing routes as well as routing performance.

6.1 Future Work

The algorithms proposed in this thesis are efficiently minimizing the communication and computation over head associated with the **Dominating Set** formation. SPA is designed to tolerate the double message loss before recomputing the markers. The results presented are simulation results considering the mobility in 2 Dimensional space without obstacles. The performance of the algorithm may be tested with 3 Dimensional space with obstacles which will be more realistic for real time implementation of Ad Hoc Wireless networks.

The connected dominating set so formed can be utilized in the Highway automation, to control congestion and improve productivity of the highways. Driver assistance programs in automobiles can be utilized more efficiently by effectively propagating congestion in the network and intensity of situations like traffic jam. The drivers can take alternative routes to avoid the situation from still being worse.

The practical implementation of Connected Dominating Set gives better estimate of its performance in real world. The algorithm may be tested using different MAC protocols and Routing protocols to evaluate its flexibility.

Bibliography

[1] Fei Dai , Jie Wu, "An Extended Localized Algorithm for Connected Dominating Set Formation in Ad Hoc Wireless Networks", *IEEE Transactions on Parallel and Distributed Systems*, v.15 n.10, pp.908-920, October 2004.

[2] J. Wu, "Extended Dominating-Set-Based Routing in Ad Hoc Wireless Networks with Unidirectional Links," *IEEE Trans. Parallel and Distributed Systems*, vol. 9, no. 3, pp. 189-200, September 2002.

[3] S.Jayadev, S.Chinara and S.K.Rath, "Weight Based Distributed Clustering Algorithm for Mobile Ad Hoc Networks", *National Symposium on Advanced Communication Technologies 2006*, Visakhapatnam, pp.199-201, Dec-2006

[4] S. Basagni, "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks", *in: Proceedings of Vehicular Technology Conference*, VTC, Vol. 2, pp. 889-893, 1999-Fall.

[5] M. Chatterjee, S.K. Das and D. Turgut, "An on-demand weighted clustering algorithm (WCA) for ad hoc networks", *in: Proceedings of IEEE GLOBECOM* 2000, San Francisco, pp. 1697- 1701, November 2000.

[6] Turgut, D., Das, S., Elmasri, R., and Turgut, B., "Optimizing clustering algorithm in mobile ad hoc networks using genetic algorithmic approach". *In IEEE GLOBECOM 2002*, 2002.

[7] C. E. Perkins and E. M. Royer. Ad Hoc On-Demand Distance Vector Routing. In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), pp. 90-100, 1999.

[8] I. Stojmenovic, M. Seddigh and J. Zunic, "Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks", *Proc. of IEEE Hawaii Int. Conf. on System Sciences* January, 2001.

[9] A.S. Tannenbaum, "Computer Networks", Third Edition, Prentice-Hall, 1996.

[10] Charles E. Perkins, "Ad Hoc Networking", Addison-wesley, New York, 2001

[11] J.Jubini and J.Tornow, "The DARPA Packet Radio Network Protocols",*Proceedings of the IEEE*, Vol. 75, No. 1, pp. 21-32, 1987.

[12] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets", Algorithmica, 20:374-387, 1998

[13] M. Gerla and J.T.C. Tsai, "Multicluster, mobile, multimedia radio network, Wireless Networks 1(3) pp. 255-265, 1995.

[14] A.K. Parekh, "Selecting routers in ad-hoc wireless networks," *in: Proceedings of the SBT/IEEE International Telecommunications Symposium*, August, 1994.

[15] S. Basagni, "Distributed clustering for ad hoc networks", *in: Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 310-315, June 1999.

[16] V. Bharghavan and B. Das, "Routing in ad hoc networks using minimum connected dominating sets", *International Conference on Communications*'97, Montreal, Canada, pp. 376-380, June, 1997.

[17] V. Chvatal, "A greedy heuristic for the set-covering problem", *Mathematics of Operation Research* 4(3), 233-235, 1979.

[18] B.N. Clark, C.J. Colbourn and D.S. Johnson, "Unit disk graphs", *Discrete Mathematics* 86, 165-177, 1990.

[19] B. Das, R. Sivakumar and V. Bharghavan, "Routing in ad-hoc networks using a spine", *Proc. of International Conference on Computers and Communications Networks* '97, Las Vegas, NV, September, 1997.

[20] R. Sivakumar, B. Das and V. Bharghavan, “An improved spine-based infrastructure for routing in ad hoc networks”, *Proc. of IEEE Symposium on Computers and Communications* ’98, Athens, Greece, June, 1998.

[21] C.R.Lin and M.Gerla, “Adaptive Clustering for Mobile Wireless Networks,” *IEEE Journal on Selected Areas in Communications*, vo;. 15, no.7 pp.1265-1275, 1997.

[22] M. Sanchez and P. Manzoni, “A java-based ad hoc networks simulator”. *Proceedings of the SCS Western Multiconference Web-based Simulation Track*, January, 1999.

[23] V. Davies, “Evaluating mobility models within an ad hoc network”. Master’s thesis, Colorado School of Mines, 2000.

[24] M. Sanchez and P. Manzoni, “Anejos: A java based simulator for ad-hoc networks”. Future Generation Computer Systems, 17(5):573-583, 2001.

[25] E. W. Weisstein, The CRC Concise Encyclopedia of Mathematics, CRC Press, 1998.

[26] D. Johnson and D. Maltz, “Dynamic source routing in ad hoc wireless networks”. In T. Imelinsky and H. Korth, editors, Mobile Computing, pages 153-181. Kluwer Academic Publishers, 1996.

[27] J. Wu and H. Li, “On calculating connected dominating set for efficient routing in ad hoc wireless networks”, *Proc. of Proceedings of the 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pp. 7-14,1999.

[28] J. Moy, OSPF version 2, RFC 2328, 1998.

[29] C. Hedrick, Routing Information Protocol, RFC 1058, 1988.

[30] C. E. Perkins and P. Bhagwat, “Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers,” Computer Communications Review, pp. 234-244, October, 1994.

[31] L. R. Ford Jr. and D. R. Fulkerson, Flows in Networks. Princeton University Press, Princeton, N.J., 1962.

[32] S. Murthy and J. J. Garcia-Luna-Aceves. “An Efficient Routing Protocol for Wireless Networks”. *ACM/Baltzer Mobile Networks and Applications*, 1(2): pp. 183-197, 1996

[33] D. B. Johnson, D. A. Maltz, Y. Hu, and J. G. Jetcheva, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)”. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-07.txt, Feb 2002.

[34] R. V. Boppana, M. K. Marina, and S. P. Konduru, “An Analysis of Routing Techniques for Mobile and Ad Hoc Networks,” *In Proceedings of International Conference on High Performance Computing (HiPC)*, pp. 239-245, 1999.

[35] J. Broch, D. Maltz, D. Johnson, Y-C. Hu, and J. Jetcheva, “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols,” *In Proceedings of IEEE/ACM MobiCom*, pp. 85-97, 1998.

[36] S. R. Das, R. Castaneda, and J. Yan, “Simulation-based Performance Evaluation of Routing Protocols for Mobile Ad hoc Networks,” ACM/Baltzer Mobile Networks and Applications (MONET), 5(3) pp. 179-189, 2000.

[37] E. Gafni and D. Bertsekas, “Distributed Algorithms for Generating Loopfree Routes in Networks with Frequently Changing Topology”, *IEEE Transactions on Communications*, 29(1) pp. 11-18, 1981.

[38] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, “Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-Hoc Networks. In Proceedings of IEEE/ACM MobiCom, pp. 195-206, 1999.

[39] V. D. Park and M. S. Corson, “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks”, *In Proceedings of IEEE Infocom*, pp. 1405-1413, 1997.

[40] C. E. Perkins, E. Belding-Royer, and S. R. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing”, http://www.ietf.org/rfc/rfc3561.txt, July 2003. RFC 3561.

[41] C. Santivanez, B. McDonald, I. Stavrakakis, and R. Ramanathan, “On the Scalability of Ad Hoc Routing Protocols,” *In Proceedings of IEEE Infocom*, pp. 1688-1697, 2002.

[42] C.-H.R. Lin and M. Gerla, "A distributed control scheme in multi-hop packet radio networks for voice/data traffic support," *in: Proceedings of IEEE GLOBECOM* pp. 1238-1242, 1995.

[43] C.-H.R. Lin and M. Gerla, "A distributed architecture for multimedia in dynamic wireless networks", *in: Proceedings of IEEE GLOBECOM* pp. 1468-1472, 1995.

[44] B. Bollbas, Random Graphs, Academic Press, 1985.

[45] R. Dube, C. D. Rais, Kuang-Yeh Wang and S. K. Tripathi. "Signal Stability-Based Adaptive Routing (SSA) for Ad-Hoc Mobile Networks," *IEEE Personal Communications*4 (1), pp.36-45, 1997.

[46] P. Jacquet, P. Muhlethaler, and A. Qayyum, "Optimized Link State Routing Protocol", IETF MANET, Internet draft, 1998.

[47] E. Royer, P.M. Melliar-Smith, and L. Moser, "An analysis of the optimum node density for ad hoc mobile networks". *In Proceedings of the IEEE International Conference on Communications (ICC)*, 2001.